INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa · University of Ottawa

University of Ottawa
Faculty of Engineering
School of Information Technology and Engineering
Ottawa-Carleton Institute for Computer Science

Feature Point Correspondences:
A Matching Constraints Survey

M.C.S. Thesis

By
Samir Arbouche

Supervised by
Dr. Robert Laganière

May 1999

Contents

Summary iv

Acknowledgments vi

List of figures vii

List of Tables x

1 Introduction 1

1.1 Calibration 1

1.2 Feature Correspondence 3

1.3 Main Contributions 5

1.4 Thesis Organization 6

2 Matching Algorithms Survey 7

2.1 Introduction 7

2.2 Graeme Framework Review 7

2.2.1 Token Types 9

2.2.2 Source of Constraints 9

2.2.3 Search Methods 10

2.3 PMF overview 12

2.4 Zhang Algorithm 12
2.5 Geometric Rigidity 13
2.6 Edgeness and Cornerness Algorithm 14
2.7 Summary 15

3 Epipolar Geometry **16**
3.1 Introduction 16
3.2 Camera Geometry 16
3.2.1 Perspective Projection 17
3.2.2 Intrinsic Parameters 18
3.2.3 Extrinsic Parameters 19
3.3 Epipolar Constraint 20
3.4 Summary 22

4 Matching Constraints **23**
4.1 Introduction 23
4.2 Points of Interest 24
4.3 Correlation Constraints 25
4.3.1 The Matching Process 25
4.3.2 Correlation 26
4.4 Confidence Measure 27
4.5 Symmetry 29
4.6 Orientation 30
4.7 Edgeness 30
4.8 Shape Similarity 31
4.9 Curvature 33
4.10 2D Rigidity 34
4.11 Figural Continuity 36

4.12 Implementation Details 38
4.12.1 Our Matching Process 39
4.12.2 A Typical Session 41
4.13 Summary 42

5 Image Test Set and Feature Points Validation **43**
5.1 Introduction 43
5.2 The Image Test Set 43
5.2.1 Source of Images 46
5.2.2 Images Calibration 47
5.3 Validating Point Correspondences 48
5.4 Summary 50

6 Summary of the Results **51**
6.1 Introduction 51
6.2 Definitions 51
6.3 Correlation Results 52
6.4 Matching Constraints Results 54
6.5 Comparison Criteria 61
6.5.1 Effectiveness 61
6.5.2 Severity 61
6.5.3 Stability 62
6.6 Overall Evaluation 62
6.7 Combining Constraints 63
6.8 Summary 65

7 Conclusion and Future Work **66**

A Graphs **68**

B Alternate graphs **79**

Bibliography **89**

Acknowledgments

I would like to thank my supervisor Dr. Robert Laganière for his great support and encouragement. He illuminated my knowledge in the field of Computer Vision and was a great help during the research process. He knew how to give me the right information when I needed it. As I would like to thank the technical staff for their support in the use of the laboratory especially Michel Racine.

List of Figures

1.1 Pinhole camera. 2

1.2 Computational framework for the matching constraint validation process. A *match set* is a set of matched pairs in a $WorkSet$ that are consistent with the applied constraint. 5

3.1 Image projection: An object in the scene is projected through the center of the camera C onto the image plane. 17

3.2 Retinal coordinate system. 18

3.3 Epipolar geometry 20

4.1 Correlation 25

4.2 Symmetry 29

4.3 Orientation 30

4.4 Normalization function for edgeness 31

4.5 Shape similarity 32

4.6 Zero-crossing patterns. 33

4.7 2D Rigidity 35

4.8 Figural continuity 37

4.9 A sample view of our application interface 40

5.1 Pairs of images used in testing. 44

5.1 Pairs of images used in testing (con't) 45

A.1 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation. 70

A.2 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the SSD correlation. 71

A.3 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and confidence measure. 72

A.4 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and orientation. 73

A.5 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and shape similarity. 74

A.6 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and edgeness. 75

A.7 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and curvature. 76

A.8 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and 2D rigidity. 77

A.9 These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and figural continuity. 78

B.1 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation. 80

B.2 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the SSD correlation. 81

B.3 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and confidence measure. 82

B.4 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and orientation. 83

B.5 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and edgeness. 84

B.6 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and Shape similarity. 85

B.7 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and curvature. 86

B.8 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and 2DRigidity. 87

B.9 This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and figural continuity. 88

List of Tables

5.1 Maximum, average and minimum disparity of all the pairs of images. . 46

5.2 Fundamental matrices associated to each image pair. 47

6.1 Initial *WorkSets* characteristics. 54

6.2 *WorkSets* characteristics before and after applying the confidence measure. Threshold is set to $th = 2$ 55

6.3 *WorkSets* characteristics before and after applying the edgeness constraint. Threshold is set to $th = 0.02$ 56

6.4 *WorkSets* characteristics before and after applying the orientation constraint. Threshold is set to $th = 0.31$ 56

6.5 *WorkSets* characteristics before and after applying the shape similarity constraint. Threshold is set to $th = 10$ 57

6.6 *WorkSets* characteristics before and after applying the curvature constraint. Threshold is set to $th = 0.1$ 58

6.7 *WorkSets* characteristics before and after applying the 2D rigidity constraint. Threshold is set to $th = 20$ 59

6.8 *WorkSets* characteristics before and after applying the figural continuity constraint. Threshold is set to $th = 75\%$ 60

6.9 *WorkSets* characteristics before and after applying the confidence measure, shape similarity and edgeness. 63

6.10 *WorkSets* characteristics before and after applying the confidence measure , shape similarity and orientation. 63

6.11 *WorkSets* characteristics before and after applying the confidence measure, shape similarity, edgeness and orientation. 64

6.12 *WorkSets* characteristics before and after applying the confidence measure, shape similarity $th = 5$. edgeness $th = 0.0001$ and orientation $th = 0.03$. 64

Chapter 1

Introduction

Many biological systems use vision as an important way of collecting information about the surrounding environment, for traveling, recognition or just appreciation. The world around us has a three-dimensional (3D) structure. When we ask someone what they see, they will have no problem to enumerate or describe objects. However, the information that is really available on each eye retina, is just a collection of two-dimensional (2D) points. Each point or pixel contains the information about the illumination and color from the scene under observation which was projected on the retina. The definition of the object is a result of an interpretation process that is part of our visual system. In computer vision field, we try to build a computational model for the visual perception.

1.1 Calibration

Geometric camera calibration is a fundamental step for any vision system and an important component of any stereo system. The camera acts as a projective transformation device that projects 3D points in a real scene onto 2D points on the retinal plane. A camera is often represented with the so-called *pinhole camera*, to illustrate

how the mapping is performed (see Figure 1.1).

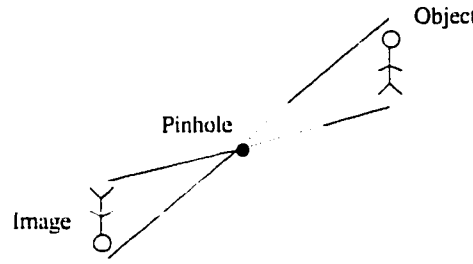


Figure 1.1: Pinhole camera.

The Calibration process then involves determining the value of intrinsic and /or extrinsic parameters[1] of the cameras as accurately as needed. These parameters give access to the camera geometry, optical characteristics, and the explicit relation between the real world coordinates and pixel coordinates in the image plane. After the calibration stage one can perform a metric reconstruction of any scene. Finally, we will be able to provide the three dimensional representation of the scene under observation.

In order to achieve this calibration, the start point is to find a set of matching objects (feature points in our case) between stereo pairs or in image sequences. This problem is well-known as the *feature correspondence problem.* Based on [Wei98], three approaches can be stated regarding the matching problem: pixel-based, area-based and feature-based. *Pixel-based* methods involve measurements at a single pixel, like the intensity or disparity, while the *area-based* methods involve the neighborhood of a pixel under the assumption that the disparity is constant in the chosen window around the pixel. In the *feature-based* methods, which are considered more advantageous because of their robustness against photometric variations, zero-crossings, gradient

[1]The projective geometry defines a matrix that maps an object onto an image, the intrinsic and extrinsic parameters constitute the projective matrix; this matter will be discussed in chapter 3 in more detail.

peaks and more structural features, are used.

The type of features someone can think of are: points of interests such as corners, edges, contours, line segments, regions, ribbons, color and complex tokens which can be constructed from simple image features like quadrilaterals from line segments.

1.2 Feature Correspondence

The feature correspondence problem can be stated as follows: *for each feature in the first image, find the corresponding feature in the other image.* These features in the images are the result of the projection of the same object in the scene. Although several solutions have been proposed in the past, finding good point correspondences between stereo pairs or in image sequences is still a very challenging task for a number of reasons:

- feature detection is not perfectly reliable, so false features may be detected in either of the images
- features in one image may be occluded in the other image.
- establishing the similarity between two features can be affected by noise in the two images.
- difference in lighting conditions, shading and reflection.
- changes in viewing position.

This is particularly true when no knowledge about the underlying camera geometry is available.

In more recent literature, the following scheme is often used to obtain image point correspondences. First, salient features are detected, most often corners [Lag98,

Har88, Smi97]. Correlation between feature points is applied and by the use of a (usually severe) threshold, an initial set of matching points is obtained. This set contains several false matches, and therefore a robust estimator (e.g., least median squares [Zha94], RANSAC [Tor97]) is performed in order to obtain the epipolar geometry of the camera system. Usually, the *epipolar constraint* (see chapter 3) reduces the search space of correspondences to one dimension along the epipolar lines if pixel points are used as features.

From this information, several false matches can be identified and rejected. Another correlation-based matching process can then be applied to obtain a new point correspondences that are in accordance with the estimated geometry of the camera system. In some cases, camera geometry is obtained by an off-line calibration step.

In such matching process, like many others, the efficiency and the accuracy of the results will be improved if one can obtain a more coherent set of corresponding points, i.e. with less false matches. Several authors have proposed different approaches in order to increase the quality of the matches. These methods use different properties and constraints[2] that can be stated about corresponding image points.
In this work, we propose to empirically compare and validate some of these methods with respect to their ability to reduce the number of false matches. The computations involved in our framework are detailed in block diagram form in Figure 1.2.

The process involved in this framework will be detailed in chapter 4. It consists of detecting the feature points to form the *InterestSets*, performing a correlation technique to obtain a set of match candidates for each point of interest (i.e. we obtain the $WorkSets$), and finally applying one of the suggested constraints to eliminate disambiguities and to form a set of possible good correspondences which has the same

[2]A constraint, in our case, consists of setting a threshold on a similarity measure score between a pair of points that form a match.

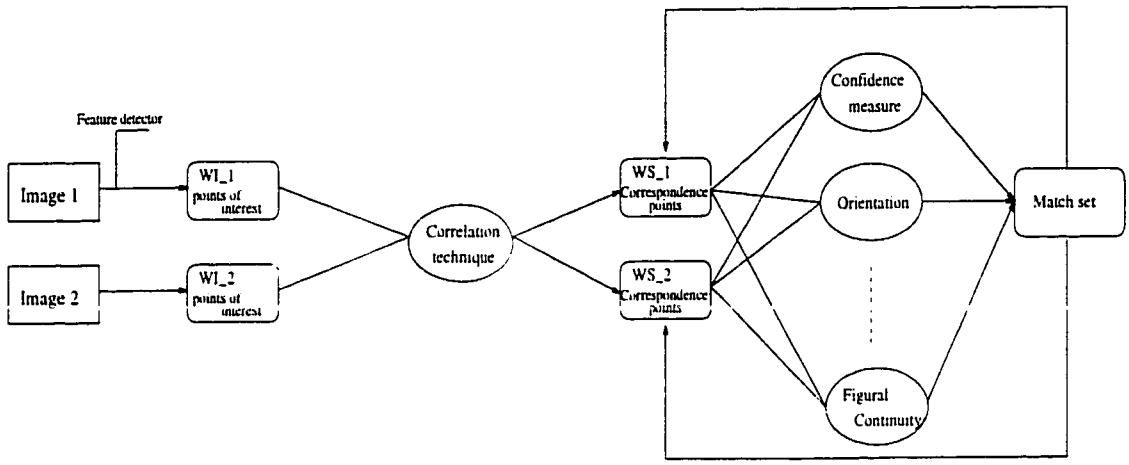


Figure 1.2: Computational framework for the matching constraint validation process. A *match set* is a set of matched pairs in a $WorkSet$ that are consistent with the applied constraint.

structure as the $WorkSets$.

1.3 Main Contributions

By reviewing existing algorithms which address matching features between images, we tried to extract some of the constraints that showed some interesting results in eliminating false matches.

Our main contribution resides primarily in investigating such constraints and empirically evaluating them on different pairs of images with different properties. The study aims to determine which ones are less or more efficient in eliminating mismatches so they can be used in new feature correspondence algorithms. This was done by developing a software that matches feature points in the images which produces an initial $WorkSet$. Each constraint was then applied separately on the $WorkSet$ to eliminate

false matches. To analyze the resulting match set, three graphs were generated which show the number of matches, the number of false matches and the distance to the epipolar line.

The second main contribution is proposing a validation process for the relevance of a constraint in detecting the false matches. It is, indeed, difficult to individually validate every point correspondence in the real image. The validation can be done visually but this is laborious work. Whereas our method, based on the epipolar geometry, permits an automatic validation using the inter-image geometry to compute the distance to the epipolar line in order to validate a match.

1.4 Thesis Organization

Chapter 2 presents a framework in which different matching algorithms can be compared to each other. It is also a survey on some of the existing algorithms that were introduced in order to build a good set of feature correspondences.

In order to have a better understanding of this work a camera geometry summary seems to be necessary. Chapter 3 reviews the camera geometry, where the camera calibration parameters are presented. Then, it introduces us to the *epipolar constraint* that will be the basis of our validation process.

Chapter 4 of this thesis describes the constraints under study and explains how each constraint will be applied on the $WorkSets$.

In chapter 5 we present the different pairs of images we investigate. This chapter also explains our approach to quantifying the validity of point correspondence sets.

Chapter 6 presents the experimental results obtained using the proposed images in chapter 5. It also concentrates on discussing and analyzing the experimental results and the performance of each constraint.

Finally, chapter 7 presents a conclusion and a discussion of future work.

Chapter 2

Matching Algorithms Survey

2.1 Introduction

The correspondence problem remains a bottleneck in computer vision. A great amount of work has been done in order to solve this problem which produced various types of algorithms. These algorithms use different approaches, matching constraints, tokens and types of search to get the match information. A very good review of these matters can be found in [Wei98, Hof87, Gra97]. Especially relevant is [Gra97], which presents a general framework by which these algorithms can be compared to each other.

A brief review of this framework is given here with some of the existing algorithms, which inspired us in selecting our matching constraints we are studying.

2.2 Graeme Framework Review

According to [Wei98], there are three different approaches which were used in common matching algorithms: area-based, pixel-based and feature-based.

Area-Based:

Area-based approaches use the absolute intensity in a neighborhood of a point in a small window centered at the point under the assumption that the disparity is constant in this region. The gray level intensity correlation between the windows of two points is then used to evaluate the match information between these related two points. Such existing work can be found in [Kim86, Gri81, Bar80, Pla97].

Pixel-Based:

Such methods use measurements at a single pixel to get the match information. The intensity differences or the disparity measurement can be used here. Various methods are described in [Pog85, Luo95, Bel96] with some variations and a formulation of refinements.

Feature-Based:

Feature-based approaches are considered more advantageous over the previous ones (area and pixel based), since, not only the absolute intensity is involved here but also variations on the intensity which provide a better characteristic of physical changes in a scene. Depending on the feature used, we can obtain rich attribute information. For example if we use line segments or edges, attributes may be such as *length*, *width*, *adjacent gray levels* and *contrast*. This approach is also faster than the area-based method [Hof87]. In the next section, we see the various features used by different algorithms.

2.2.1 Token Types

The type of tokens used in the literature are: zero-crossings [Kim87], gradient peaks [Wen], edges [Gri81, Kim86] and points of interest[1] (like in our case). Other more complex tokens have also been used, such as, line segments, contours, regions and even colors. These matching primitives allow working with richer attribute information. However the computation in this case is more expensive.

2.2.2 Source of Constraints

A detailed description will be given in chapter 4 on constraints which are classified in two categories:

Unary Constraints

Unary matching constraint sets homologous features based on the assumption that the attributes are similar in the neighborhood of a point. Among which we find the epipolar constraint (geometric) [Zha94, Fau96, HoM95]. The second major source of unary match information is similarity constraints which are based on token attributes. They allow to further disambiguate the matches: DGL (disparity gradient limit) [Pol85, HoM95], zero-crossing pattern matching [Kim87], WFP windowed Fourier phase (using zero crossing and peaks) [Wen], edgeness and cornerness constraints [Wen98], zero-crossing of the Laplacian of a Gaussian (same sign and contour orientation) [Jor91], disparity similarity [Hu94, Pra85], geometric and rigidity constraints [Hu94]... etc.

[1] corners, points on edges.

Compatibility Constraints

Compatibility or binary constraints ensure that two pairs of matches are mutually compatible. For example:
Continuity (figural continuity [Hof87, HoM95]): requires that points adjacent in 3D space remain adjacent in each image.
Topology: orderness [HoM95, Fau96] is one of them, 3D structure viewed in both images is identical which implies that the projection of this structure should reflect this constraint in different viewpoints. Topology preserves parallelism, relative position, connectivity and collinearity.
N-ary constraints are also available.

2.2.3 Search Methods

There does not exist a general method to extract a final set of matches. The simplest method results in accepting matches based on the similarity of matched tokens attributes being greater than a certain threshold. In general, a search method is characterized by a criterion to decide whether a match is correct or not and a search algorithm. Among such algorithms, we have:

Hierarchical Search

A hierarchy of different resolution levels is used to represent the images, from a high to a low resolution. The match information obtained at a low resolution level is then used to obtain the correspondences at the higher resolution. It is a top-down search method [Han84, Han89] where the top level corresponds to the lowest resolution of the images.

Dynamic Programming

In this method, points along two correspondent epipolar lines are matched in a unique way by dynamic programming. The epipolar, unicity and orderness constraints are imposed here. Most of the techniques use an optimization search method. The function to optimize is based on the sum of match (dis)similarities. Other than epipolar, unicity and orderness constraints, we can also take into account the figural continuity [Oht85]and DGL. Other implementations can be found in [Bak81, Li94].

Relaxation Method

The epipolar constraint is exploited in this case combined with a similarity measure to obtain an initial set of correspondences. A relaxation algorithm is then applied to select matches that are consistent with the unicity and DGL constraints. A common algorithm which uses this method is the PMF algorithm [Jor91]. The relaxation method consists of:

- a set of features (points, edges, surfaces) belonging to an object and a set of labels.
- for each feature, weights (confidence measure) or probabilities are assigned to each label in the set to produce an estimate of support that the particular label is the correct one for that feature.
- Probabilistic approaches are then used to maximize or minimize the probabilities by iterative adjustment, taking into account the probabilities associated with features in the neighborhood.

Note the figural continuity can also be exploited during the iteration process.

Matching as Graph Isomorphism

Every image is represented as a graph. The correspondence problem is then reduced to a graph correspondence problem. This approach was introduced in [Sko88, Hor87].

2.3 PMF overview

An overview of this algorithm is also presented in [Jor91]. The algorithm is divided into two steps:

- Initialization: given a pair (m, m') which is consistent with the epipolar constraint, the confidence measure or match support $S(m, m')$ is computed by the equation 4.8 given in section 4.4.
- Iteration: m' is a match for m if $S(m, m')$ is maximum in the vicinity of m. In a second step, we preserve only the pairs for which $S(m, m') = S(m', m)$. The third step consists of applying the uniqueness constraint. All consistent pairs are eliminated during the iteration process.

A candidate match (m, m') is then retained, eliminated or not treated. The algorithm stops when no untreated matches exist anymore.

Obviously, this algorithm is based on the epipolar constraint and the uniqueness constraint. Figural continuity could also be explored here. The search method used is the relaxation labeling.

2.4 Zhang Algorithm

This algorithm [Zha94] is based on the epipolar constraint to establish a correspondence set. First, a classical method is used to get an initial set of matches. A *correlation technique* and relaxation methods were used followed by a robust technique, the

Least Median of Squares, to reduce the false matches. The epipolar geometry is then computed by recovering the fundamental matrix. Finally, the set of correspondences is established by using stereo matching[2] and the estimated camera geometry.

From Zhang algorithm, and since it seems to give encouraging results, we were inspired for the use of the VNC correlation and also the confidence measure.

2.5 Geometric Rigidity

This algorithm was proposed in 1994 by Xiaoping Hu and Narendra Ahuja [Hu94] and is based on a combination of intensity information and geometric constraints. As in any other algorithm, a set of feature points is first detected by applying any feature detector algorithm (corner detector algorithm). Potential candidate matches are then established based on the similarity of local intensities. The algorithm attempts to find a set of correspondence points that are locally intensity-matched and do preserve a geometrical structure. The five steps in the algorithm are given by:

1. Point feature detection

2. Intensity-based matching

3. Geometry-based elimination: three tests are involved here. *Test A* requires that the triangle made by a point and its 2 closest neighbors in one image are similar to that formed by their matches in the other image. All the matches are in S_r, the set of rigid points that passed *test A*. This test is applied repeatedly until no more good matches exist. In *test B*, we check for each point[3] in S_r if there exist two closest neighbors that do preserve the 2D rigidity test. Finally, *test C*,

[2]Establishing a new correspondence set using a correlation technique that takes into account the epipolar geometry.

[3]this point might have been tagged as non rigid in *test A*.

requires that when we choose the nearest neighbor we select the one that has the same disparity.

4. Rigidity-based elimination: this is a 3D geometric test. In this case, knowledge of the depth is required. We test if each 3D-triangle in the first image has the same size of its corresponding 3D-triangle in the second image.

5. Disparity-based elimination: to ensure that no matched feature point in an image could be rematched with another feature point that could have another disparity value associated with.

The results obtained by this algorithm presented in [Hu94] encourage us to explore the idea of the 2D rigidity test, since the other tests require knowledge of the depth which we are avoiding in our case.

2.6 Edgeness and Cornerness Algorithm

Attributes such intensity, edgeness (high transition of intensity) and cornerness are used together in [Wen98] for the matching process. The edgeness is based on the gradient magnitude which is normalized. Cornerness is also normalized and is a function of the edgeness. Cornerness measures the changes of the gradient at two points in the vicinity of the considered point where the cornerness is to be computed. The formulas can be found in [Wen98]. Other criteria, besides the ones enumerated above, are used by this algorithm: Intra-Regional Smoothness in the case where the intensity variation has no significance and the Occlusion criterion.

In this algorithm a hierarchical search was used to get the match information. But instead of blurring only the intensity image from a finer level to a coarser one, original edgeness and cornerness images were also blurred. The matching problem is reduced to an optimization problem where we want to minimize a weighted sum of squares

of residual errors that result from similarity measures of all matching constraints: intensity, edgeness, cornerness, orientation smoothness and displacement smoothness.

2.7 Summary

This chapter summarized a general framework of the different algorithms used to solve the correspondence problem. The framework suggests that they can be categorized according to the approach, features, constraints and type of search used to get the match information. In a second part we gave an overview of some of the existing algorithms. These algorithms inspired us in choosing the constraints under study. The correlation and confidence measure constraints from Zhang algorithm, edgeness and shape similarity constraints from [Wen98, Kim87], 2D rigidity constraint [Hu94] and figural continuity constraint [HoM95].

Chapter 3

Epipolar Geometry

3.1 Introduction

In this chapter, we present a brief review of the geometric model associated with a camera system. This model is characterized by a number of parameters that are estimated in the calibration process. We will then define these intrinsic and extrinsic parameters and introduce you to the *epipolar constraint*.

3.2 Camera Geometry

Under the pinhole model, the camera performs a perspective projection of a 3D point onto a pixel point located on the retinal plane (Figure 3.1). This projection is performed in two transformation steps:

1. A projection that transforms a 3D point onto a 2D point in the image plane.

2. A transformation from the image plane into a metric coordinate system associated with the retinal plane.

3.2.1 Perspective Projection

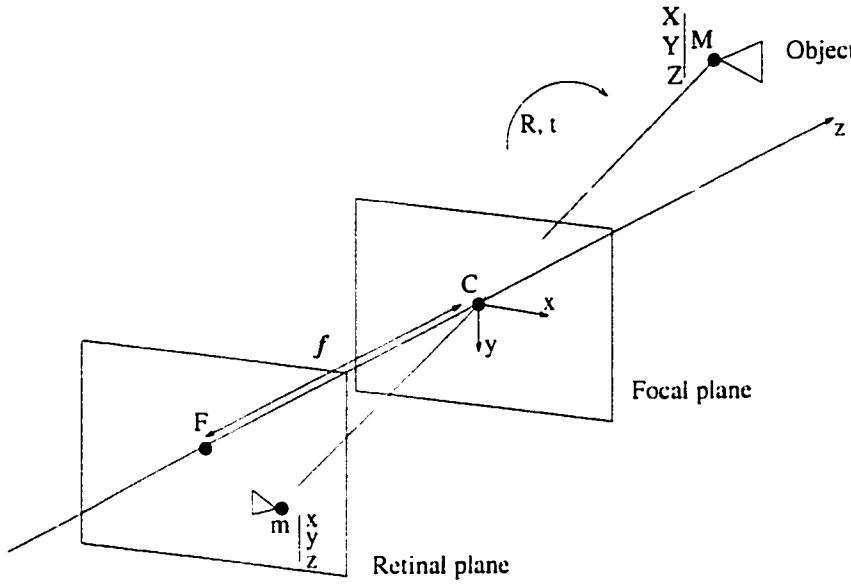


Figure 3.1: Image projection: An object in the scene is projected through the center of the camera C onto the image plane.

Using homogeneous coordinates, where a 3D point is written as $\tilde{\mathbf{M}} = [X\ Y\ Z\ 1]^T$ and its image point as $\tilde{\mathbf{m}} = [x\ y\ 1]^T$, the relationship between the world coordinates and the pixel coordinates becomes linear projective:

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \tag{3.1}$$

where the equality is up to a scale factor s. $\mathbf{P}$ is the **projective matrix** and can be decomposed as:

$$\mathbf{P} = \mathbf{A}[\mathbf{R}\mathbf{t}] \tag{3.2}$$

where $\mathbf{A}$ is a 3×3 matrix transforming the image coordinates to the retinal image coordinates, $(\mathbf{R}, \mathbf{t})$ is the displacement between the world coordinate system and the

camera coordinate system. **A** is obtained as follows: as a result of the similarity between the two triangles ($\mathbf{MM_zC}$) (where $\mathbf{M_z}$ is the projection of **M** on the **z** axis) and (**mFC**) (Figure 3.1) we have the following relationship:

$$\frac{f}{Z} = \frac{x}{X} = \frac{y}{Y} \tag{3.3}$$

and a point $\mathbf{M}(X, Y, Z)$ is projected onto the retinal plane at the point $\mathbf{m}(x, y)$ according to the mapping:

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.4}$$

3.2.2 Intrinsic Parameters

The retinal plane has its own coordinate system and (Equation 3.4) is redefined as the transformation (scaling and rotation):

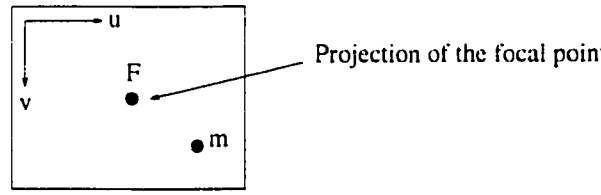


Figure 3.2: Retinal coordinate system.

$$\begin{cases} u = \frac{-fX}{l_x Z} + u_0 \\ v = \frac{-fY}{l_y Z} + v_0 \end{cases} \tag{3.5}$$

where l_x and l_y are the dimensions of the pixel in the world coordinate unit, (u, v) is **m** expressed in the retinal coordinate system.

Note that the retinal plane could have been before the focal plane, the result would be the same but for a sign difference in (equation 3.5).
We thus have four intrinsic parameters:

- l_x *and* l_y are the horizontal and vertical scale factors.
- (u_0, v_0) are the coordinates of the principal point of the camera.
- f: focal length of the camera.
- α_u *and* α_v: $(\alpha_u = \frac{f}{l_x})$, $(\alpha_v = \frac{f}{l_y})$. It can be interpreted as the focal length in both directions of the camera system expressed in pixels.

Another parameter (angle θ) can be taken into consideration. It expresses the fact that the pixel grid may have some distortion. But in practice this angle is close to $\pi/2$. Equation 3.4 becomes then, in its homogeneous representation:

$$s\tilde{\mathbf{m}} = \mathbf{A}\tilde{\mathbf{M}} \tag{3.6}$$

where $\mathbf{A}$ is called the **intrinsic matrix**

$$\mathbf{A} = \begin{pmatrix} -\alpha_u & 0 & u_0 & 0 \\ 0 & -\alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.7}$$

3.2.3 Extrinsic Parameters

The motion between the camera and the object, is expressed by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{3.8}$$

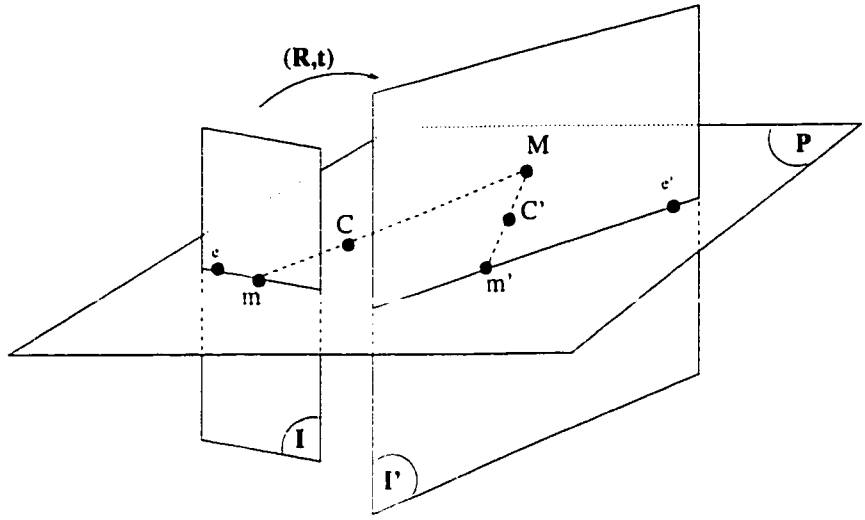


Figure 3.3: Epipolar geometry

$$s\tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R}.\,\mathbf{t}]\tilde{\mathbf{M}} \tag{3.9}$$

We have 3 rotations and 3 translations about the three world coordinate axes. These are the **6 extrinsic parameters**.

3.3 Epipolar Constraint

When two such cameras are considered, the underlying geometry gives rise to a relation between corresponding image points, namely $\mathbf{m}$ and $\mathbf{m}'$ (images of the same 3D point), which can be expressed as:

$$\ell' = \mathbf{F}\mathbf{m} \tag{3.10}$$

ℓ' is the equation of a line formed by $(\mathbf{m}'\mathbf{e}')$, called the epipolar line, on which the corresponding point $\mathbf{m}'$ must lie. $\mathbf{F}$ is the fundamental matrix that contains all the geometric information relating the two views. Since a similar relation also holds for

point $\mathbf{m}'$ and its corresponding epipolar line $\ell(\mathbf{me})$, we finally obtain the following equation that must be true for any pair of corresponding points:

$$\mathbf{m}'^T\mathbf{F}\mathbf{m} = \mathbf{0} \tag{3.11}$$

it defines the *epipolar constraint.* It will be the basis of our validation process regarding false matches. Indeed to be considered as a good match, a pair $(\mathbf{m}, \mathbf{m}')$ should satisfy the fact that $\mathbf{m}'$ lies on the epipolar line related to $\mathbf{m}$.

Equation 3.11 can be shown by the following:
Assuming that the world coordinate system is situated at the first image center and that the displacement between the two cameras is given by $(\mathbf{R}, \mathbf{t})$, we have the following two equations:

$$s\tilde{\mathbf{m}} = \mathbf{A}\,[\mathbf{I}, 0]\,\tilde{\mathbf{M}} \quad s'\tilde{\mathbf{m}}' = \mathbf{A}'\,[\mathbf{R}, \mathbf{t}]\,\tilde{\mathbf{M}} \tag{3.12}$$

where $\mathbf{A}$ and $\mathbf{A}'$ are the intrinsic matrices of the two images. By eliminating $\mathbf{M}$, $\mathbf{s}$ and $\mathbf{s}'$ from the above equation, we obtain :

$$\tilde{\mathbf{m}}'^T \underbrace{\mathbf{A}'^{-T}\mathbf{T}\mathbf{R}\mathbf{A}^{-1}}\,\tilde{\mathbf{m}} = 0 \tag{3.13}$$

$$\mathbf{F} = \mathbf{A}'^{-T}\mathbf{T}\mathbf{R}\mathbf{A}^{-1} \tag{3.14}$$

$\mathbf{F}$ is the **fundamental matrix** of the two images. $\mathbf{T}$ is the matrix defined by the translation vector $\mathbf{t}$:

$$\begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \tag{3.15}$$

3.4 Summary

We have presented, in this chapter, a review of the camera geometry based on the perspective projection. In the second section we introduced the parameters to estimate in a geometric calibration process (intrinsic and extrinsic parameters). In the last section we defined the epipolar constraint that relates two correspondence points if they are perspective projections of one and the same object point in the scene.

Chapter 4

Matching Constraints

4.1 Introduction

In the literature, a large number of algorithms attempted to solve the difficult problem of feature correspondence. The implementation of these algorithms differs in the token type, matching constraints and search type used to determine the match information. [Gra97, Wei98] illustrate well this matter.

We can find different types of matching constraints in the literature, first we have correlation constraints, which constitute the initial stage of a matching process since for every extracted feature in the first image, such type of constraints allows the detection of potential correspondences in the second image. This technique is mostly intensity-based. Another category is geometric constraints, such as the epipolar constraint, figural continuity and two-dimensional or three-dimensional rigidity. Physical constraints can also be used to reduce mismatches, because the ordering of homologous points along the epipolar must be the same in both images. The other important family of constraints on stereo images or image sequences uses features associated with

the edge detection. Which are gradient-based where the first derivative of the intensity is involved or second order constraint where the features correspond to zero-crossing points. Similarity measures can also be viewed as another type of constraints, such as disparity and intensity similarity measurements or any other similarity measurement [Pra85]. Confidence measures can also be useful an asset too in eliminating false matches. Our results show its efficiency.
In the following section we introduce some of the above constraints and explain how the matching process is performed as shown in the framework of Figure 1.2.

4.2 Points of Interest

First, we need to extract a set of points of interest between the two frames, we call them IS for one image, and IS' for the other image. The feature points may correspond to high curvature points and can be computed by applying a corner detector algorithm or/and an edge detector algorithm independently to the two images. In our case, we used the corner detector described in [Lag98]. In order to reduce the number of feature points for which the detection is due to noise, or to some irrelevant high frequency patterns, we retained only feature points located on the edges detected by the *Canny* edge detector [Can86].

Once we get IS and IS' the *InterestSets*, we can start the matching process by applying a correlation technique, to obtain a set of candidates for each point in IS and in IS' in the opposite image. The new obtained sets are called the $WorkSets$.
Finally, after establishing an initial list of potential correspondence points, a series of constraints is applied in both directions (left-right image, right-left image) in order to reject the possible false matches. At this point, we only retain in the $WorkSets$ the matches that remain consistent, so far, with the applied constraint.

4.3 Correlation Constraints

4.3.1 The Matching Process

The image matching process is performed by searching for pixel correspondences in a predefined window of the second image. The correlation score is obtained by performing either a hierarchical search or a computation at one single resolution. In our work we explore linear correlation methods, found in the second category. Typical measures to compute the correlation score between a pair of potential candidates are presented in the following papers [Han74, Zha94, Din98, Ana84, Dar98, Zha93]. Such measures are the direct correlation, mean normalized correlation (NCC), variance normalized correlation (VNC), sum of squares of the differences between the corresponding pixels (SSD), and sum of magnitudes of the differences (SMD). In our case, we explore VNC and SSD. While all correlation techniques named above are intensity-based, [Din98] introduced an ordinal correlation measure. Rather than the absolute intensity, the intensity is considered as an ordinal variable. The correlation score is then equal to the distance between two rank permutations.

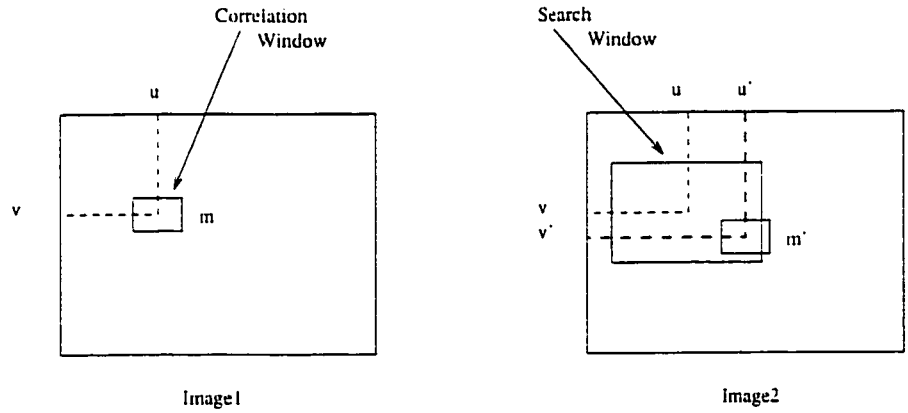


Figure 4.1: Correlation

Given a high curvature point $m(u, v)$ in the first image, we consider a correlation window of size $(2n + 1) \times (2p + 1)$ centered at point m. In the second image and around the location (u, v), we search in a rectangular area of size $(dmaxU - dminU) \times (dmaxV - dminV)$, and perform a correlation operation between the correlation window in the first image and all points of interest m' within the search area in the second image. To get the putative corresponding points, a threshold th on the correlation score is set. The pair of candidates with a score higher than th will are added to the *WorkSet* WS. The so-called *WorkSet* is defined as follows:

$$WS = \{(m, m') : m \in IS \text{ and } m' \in IS' \text{ where Score(m, m')} \geq th\}$$

Note that if $(m, m') \in WS$ then $(m', m) \in WS'$. This property does not hold true when other non-symmetrical constraints are applied.

4.3.2 Correlation

In our work we explore two commonly used correlation functions: variance normalized correlation (VNC) and sum of squares of differences between corresponding pixels (SSD). If $m(u, v)$ and $m'(u', v')$ is the candidate match, then VNC is defined as:

$$Score(m, m') = \frac{\sum_{i=-n}^{n} \sum_{j=-p}^{p} [I(u + i, v + j) - \overline{I(u, v)}][I'(u' + i, v' + j) - \overline{I'(u', v')}]}{(2n + 1)(2p + 1)\sqrt{\sigma^2(I)\sigma^2(I')}} \tag{4.1}$$

where

$$\overline{I(u, v)} = \frac{\sum_{i=-n}^{n} \sum_{j=-p}^{p} I(u + i, v + j)}{(2n + 1)(2p + 1)} \tag{4.2}$$

represents the intensity mean over the correlation window and $\sigma(I)$ is the standard deviation of I which is given by:

$$\sigma(I) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-p}^{p}(I(u+i,v+j) - \overline{I(u,v)})^2}{(2n+1)(2p+1)}} \tag{4.3}$$

$Score(m, m')$ is in [-1,1]. The score is 1 when the correlation windows are identical.

SSD is given by the following formula:

$$Score(m, m') = \sum_{i=-n}^{n}\sum_{j=-p}^{p}[I(u+i,v+j) - I'(u'+i,v'+j)]^2 \tag{4.4}$$

In conclusion, for every point in the first image, we have a set of candidate points from the second image (possibly nil). Moreover, if the constraint on the correlation score is satisfied in one direction it remains consistent also in the other direction conserving the score symmetry.

In our experiments, we set $n = p = 3$ and search area close to the quarter of the image area.

4.4 Confidence Measure

In order to reduce the number of false matches, a confidence measure has been proposed to estimate the consistency of a match with respect to its neighbors. It is based on the so-called *strength of the match* [Zha94], and is defined as follows:

$$sm(m_i, m'_j) = c_{ij}\sum_{n_k \in N(m_i)}[\max_{n'_l \in N(m'_j)}\frac{2c_{kl}\delta(m_i, m'_j; n_k, n'_l)}{2 + d(m_i, n_k) + d(m'_j, n'_l)}] \tag{4.5}$$

c_{ij} and c_{kl} are the correlation scores previously computed, however be any similarity measure which represents the goodness of the candidate matches, can be used in *sm*. d(...) is the Euclidean distance and :

$$\delta(m_i, m'_j; n_k, n'_l) = \begin{cases} e^{-r/\varepsilon_r} & \text{if}(n_k, n'_l) \text{ is a candidate match and } r < \varepsilon_r \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

r is the relative distance difference:

$$\frac{2\left|d(m_i, n_k) - d(m'_j, n'_l)\right|}{d(m_i, n_k) + d(m'_j, n'_l)} \tag{4.7}$$

where ε_r is a threshold on r. $N(m_i)$ is the neighbors set of m_i. If VNC is applied, equation (4.5) counts the number of candidate matches found in the neighborhood of the considered match, where the relative distance between their positions is similar. In general, this measure is not symmetric, i.e. $sm(m_i, m'_j) \neq sm(m'_j, m_i)$.

The confidence measure defined above is very similar to that used by the PMF algorithm [HoM95, Pol85](see section 2.3 also):

$$S(m, m') = \sum_{i \in N(m)} [\max_{j' \in N(m')} \frac{C(i, j')G(m, m', i, j')}{d(m, i)}] \tag{4.8}$$

- In this expression $N(.)$ represents a neighborhood of (.). $C(i, j')$ is the correlation score:

$$C(m, m') = \sum_{i=-n}^{n} \sum_{j=-p}^{p} [I(u + i, v + j)I'(u' + i, v' + j)] \tag{4.9}$$

- $d(m, i)$ is a proportional function to the distance between m and i.

- G is the disparity gradient support associated with matches (m, m') and (i, j'), see [HoM95].

In addition [Ana84] presented another type of confidence measure that deals with homogeneous and occluded areas to improve the accuracy of the correlation matching algorithm. This confidence measure is based on the *SSD Surface*. The *SSD Surface* is formed by considering the Laplacian-filtered SSD values of different candidate points in a window centered at the best match. Usually at this point (best match) there are significant intensity variations, in addition the SSD surface has the property that the

SSD value at the best match seems to be low. It is also found that the curvature of the SSD surface along different directions reflects the degree of intensity variation in the image along those directions. The score is then given by:

$$min(C0.C45, C90, C135) \tag{4.10}$$

where $(C0, C45, C90, C135)$ are the normalized second derivatives of the SSD surface centered at the best match point in the four directions 0, 45, 90 and 135 degrees.

4.5 Symmetry

This constraint, simply, imposes a symmetry between the two $WorkSetsWS$ and WS'. That is, if a point m has a candidate match m', $(m,m')\in WS$, then m must be in the list of candidates of m', $(m'.m) \in WS'$. Any pair that does not satisfy this symmetry is eliminated. This constraint can be used only after a non-symmetrical constraint has been applied.

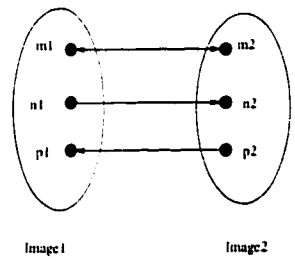


Figure 4.2: Symmetry

The pairings (n_1, n_2) and (p_1, p_2) are rejected in this example. The arrows indicate the direction of the matching pair being consistent with the constraint applied so far.

4.6 Orientation

The similarity between the intensity gradient orientation at a point (u, v) and its corresponding point is involved here. It is about the angle between the director vector associated with this point (in the case of edge feature points, it is the tangent to the edge at this location) and one of the image axes. The angle value is computed up to 2π.

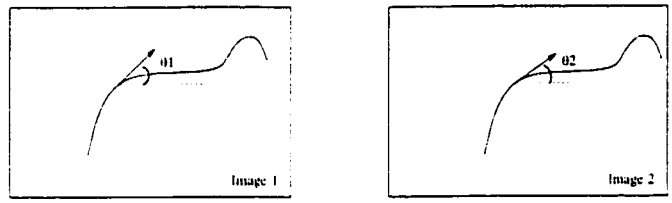


Figure 4.3: Orientation

For each pair of candidate matches, to remain consistent with the orientation constraint, the square of the difference between the angles made by the gradient vectors and the horizontal axis are tested with the imposed threshold th to determine if it is a false match. Let's say $m(u, v)$ corresponds to $m'(u', v')$,the similarity measure is given by:

$$o(m, m') = [\arctan(I_u/I_v) - \arctan(I'_u/I'_v)]^2 \tag{4.11}$$

We just make sure that the angles cover all the circle, when computing the *arctan* function. In our experiments, we tolerated an angle difference between $[5^o, 45^o]$.

4.7 Edgeness

Based on the work done in [Wen98], the intensity images are filtered to suppress gray level noise and may be scaled linearly such that the minimum and maximum values

of the intensity are 0 and 255, respectively. The edgeness measure is defined by:

$$e(m) = f(\|\nabla I(m)\|) \tag{4.12}$$

where f is the normalization function shown in Figure 4.7 with a two transition points x_0 and x_1. $\nabla I(m) = [I_x, I_y]^T$ is the gradient of the image intensity at a point m.

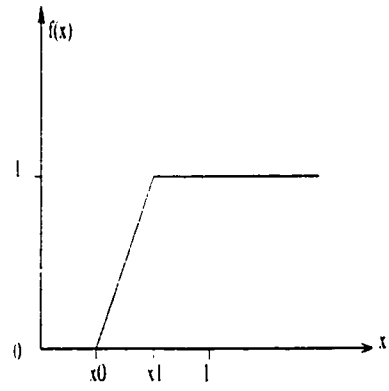


Figure 4.4: Normalization function for edgeness

The edgeness is defined to range from 0 to 1 with smooth transition between x_0 and x_1. In the equation (4.12), x_0 and x_1 are represented by $\|\nabla I(m_0)\|$ and $\|\nabla I(m_1)\|$ respectively. The value of these parameters can be determined by analyzing gradient magnitudes. However, these parameters can also be entered manually in the algorithm implementation. We conclude that if $se(m, m') = (e(m) - e(m'))^2 < \epsilon$ for $\epsilon > 0$ then the match is consistent otherwise it is rejected from the $WorkSets$.

4.8 Shape Similarity

For each pairing (m, m') in the $WorkSet$ WS, we are trying to detect mismatches based on similarity in shapes of corresponding corners. The constraint is performed as follows: a circular window $W(.)$, with radius r, is extracted at every location m

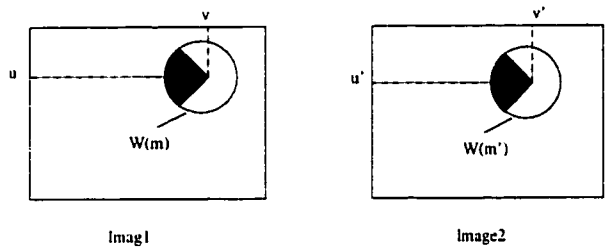


Figure 4.5: Shape similarity

and m'. The mean of the intensity, $\overline{I_{W(m)}}$ and $\overline{I_{W(m')}}$, inside these windows are then computed. Each point (i, j) of the circular windows is transformed according to the following formula:

$$\rho(I(i,j)) = \begin{cases} 1 & \text{if } I(i,j) > \overline{I_{W(m)}} \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

The dark portions of the windows $W(m)$ and $W(m')$ (Figure 4.8) represent points with gray level intensity bigger than the overall intensity mean.

This transformation can be seen as a rudimentary Block Truncation Coding [Del79]. The similarity measure between m and m' is simply the Hamming distance between the resulting windows. i.e the number of corresponding pixels that do not have equal values.

This idea of shape similarity is also explored in [Kim87]. In [Kim87], zero-crossing points are associated with one of the 16 possible zero-crossing patterns (Figure 4.8) according to their local connectivity. A zero-crossing point in the first image is then matched with a zero-crossing point in the second image according to a pattern measure function which defines their similarity.

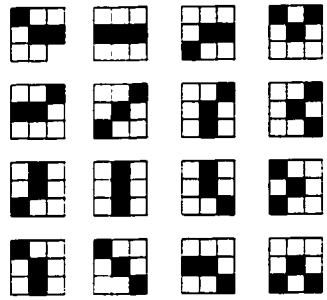


Figure 4.6: Zero-crossing patterns.

4.9 Curvature

If the gradient at a point, which corresponds to the surface normal, provides information about the surface; we would like to also explore the curvature degree at this point. This leads us to the *curvature constraint.* In fact, the curvature is defined by the local normal orientation variation which can be expressed by the second derivative of the image intensity.

Here are the steps to applying curvature constraint:

1. Like the orientation, the intensity gradient $\nabla I = (I_u, I_v)$ is computed for both images.

2. A Gaussian filter is applied to the images. The smoothing is performed by convolving an image with a *Gaussian operator* which is defined by:

$$g(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \tag{4.14}$$

   where σ is the parameter to control the gray level noise. σ is the standard deviation and represents the width of the *Gaussian distribution.*

3. Having smoothened the images with a Gaussian operator we compute the Laplacian of the smoothed image. The Laplacian is the result of the image convolution

by the following mask:

1	1	1
1	-8	1
1	1	1

4. The last step is to compute the curvatureness or the degree of the curvature at a point m. The edge curvature as defined in [Fau96] is given by:

$$\kappa(m) = \frac{-\nabla^2 I(m)}{\|\nabla I(m)\|} \tag{4.15}$$

An alternative way to compute $\kappa(m)$ is to use partial derivatives as obtained by applying the Haralick-like filters defined in [Vie92]. But we found that this approach shows increased instability.

Again, we define the similarity measure for two points in the first and second image as the square difference between the two measurements:

$$s\kappa(m, m') = (\kappa(m) - \kappa(m'))^2 \tag{4.16}$$

if the points does not satisfy this similarity constraint, they are rejected from the $WorkSet$.

4.10 2D Rigidity

This is a geometric constraint based on the work of [Hu94]. The aim is to identify and remove points that do not preserve the two dimensional consistency in the image plane.

Triplets of matched tuples are considered simultaneously to test the *2D rigidity*, see Figure 4.7.

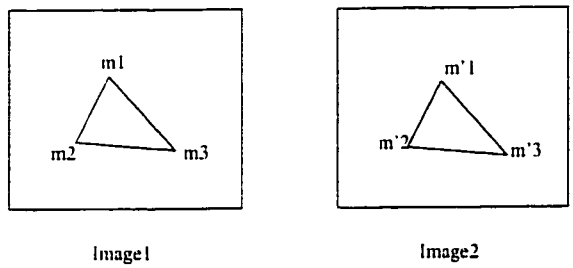


Figure 4.7: 2D Rigidity

Initially, all the pairs in the $WorkSet$ WS are considered as rejected. For all candidate matched pairs (m_i, m'_j) in WS, we search for the two closest neighbors of m_i namely, m_k and m_l within a disk of radius R in the first image. For these points there exist m'_r and m'_s, where (m_k, m'_r) and (m_l, m'_s) belong to WS. Besides that, the disparity at m_k and m_l should be nearly similar to that at m_i. Once we determine the triplets: (m_i, m_k, m_l) and (m'_j, m'_r, m'_s), and if the triangle similarity between the two triangles $(m_i, \widehat{m_k}, m_l)$ and $(m'_j, \widehat{m'_r}, m'_s)$ holds then all the three matched pairs become consistent with the constraint, and are tagged as *rigid*, otherwise no change is made. By the end of the process, we only keep in WS and WS' pairs with a rigid tag. In conclusion, only when the above conditions are satisfied, can a mismatched pair m_i, m'_j survive the two dimensional rigidity test. Note: (m_k, m'_r) and (m_l, m'_s) from above are also rigid.

Triangle Similarity:

In Figure 2.6, triangle similarity measure is expressed by:

$$str[(m_1, \widehat{m_2}, m_3), (m'_1, \widehat{m'_2}, m'_3)] = \{[d(m_1, m_2) - d(m'_1, m'_2)]^2 + [d(m_2, m_3) - d(m'_2, m'_3)]^2 + [d(m_3, m_1) - d(m'_3, m'_1)]^2\} \tag{4.17}$$

which represents the proportions of the lengths of the corresponding sides in the triangles.

Disparity Similarity:

The disparity similarity measurement used in this algorithm is the euclidean distance, in pixels, between two points.

For instance, the constraint, presented here, preserves the local geometric consistency in the two dimensional image plane. If one could examine the rigidity in the 3D-space, such that all matched pairs correspond to the same scene point, the resulting correspondence subset would be more accurate. But because of the lack of the depth information that requires the epipolar geometry, we do not explore this option.

4.11 Figural Continuity

The figural continuity constraint, as described in [HoM95], suggests that points on an edge in the first image should be matched with points in the second image on the same edge. However this approach is difficult to implement. Points on an edge in the first image have a certain order which can be lost on the edge containing the corresponding points in the second image. Corresponding points should have the same order. Also, in the case where corresponding points are found on different but intersecting edges, it is hard to tell which edge is going to be considered as the similar edge or do we consider them as one edge?

In our implementation, we consider all feature points located on a given edge. If a substantial number of these points is matched with a number of feature points lying on a common edge in the other image, then such points are considered to be consistent with respect to the figural continuity constraint. This idea is illustrated

in figure 4.8 where a majority of feature points on the curve, labeled 3, are matched with points on curve 5 in the other image. If the percentage of the matched points on curve 3 is higher than a given threshold then the points on curve 5 are accepted. In this example the two feature points that match with points on curve 9 and the one matched with a point on curve 2 would be rejected.

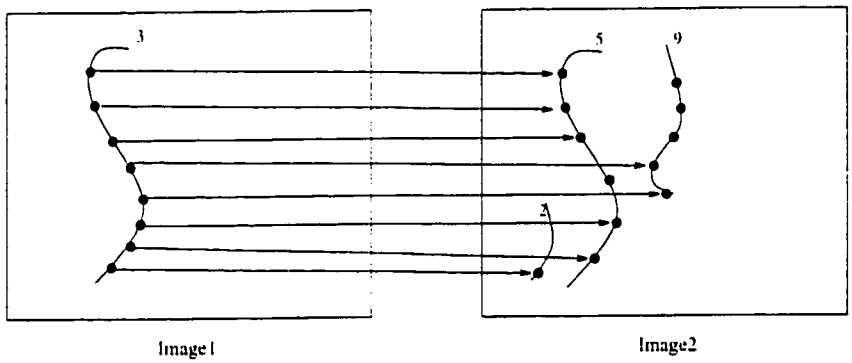


Figure 4.8: Figural continuity

The steps to perform our figural continuity algorithm are given by the following:

- Each point m_i in the left image and its match m'_j in the right image are respectively associated with an edge in their corresponding frame. Edges are numbered from 1 to max and 1 to max' respectively for both images. We denote $No_e(m_i)$ as the function which returns the index of the edge containing m_i.

- The number of points on every edge are counted and stored in an edge vector e for the first image, for example.

index	1	2	3	4
e	5	7	4	...

- A matrix M of size $max \times max'$, can be used, such that:
  whenever a match (m_i, m'_j) exists in WS the value of $M(No_{e'}(m'_j), No_{e'}(m'_i))$ is incremented. For each edge in the left image we will have the number of points

in the right image which are involved in a match with edge points, on all edges in the right image:

	r_1	r_2	...	$r_{m'}$
	5	0	...	4
l_2	0	0	...	6
...	3	0	...	0
l_m	5	0	...	2

where $r_i = l_i = i$, r for right image and l for left image.

- To eliminate disambiguities, we set a threshold $th = 65\%$, and for all candidate matches in WS_1, only those where $M(Noe'(m'_j), Noe'(m_i))$ is greater than th remain consistent with our figural continuity constraint.

4.12 Implementation Details

Our matching process algorithm is similar to the *bi-partite graph matching* algorithms. However there is a slight difference between our matching algorithm and the bi-partite matching algorithms. In the bi-partite graph matching we, usually, look for maximum matching or perfect matching[1] in a graph $G = (A \cup B, E)$ where $A \cup B$ is the set of nodes and E is the set of edges connecting nodes in A to nodes in B only. Whereas in our case, an initial set of matches is obtained using a correlation technique and each constraint is applied to eliminate some of the edges which correspond to false matches.

Every image corresponds to a set of points (feature points) with numerous information associated with it (attributes). The information can be stated as gray level intensity, intensity gradient, gradient magnitude, etc... We assume that sets A and B represent the two images respectively. Points in set A are then connected to points

[1]The bi-partite graph matching is not limited to this two algorithms only.

in set B by edges which have weights associated with them. These weights constitute correlation or matching constraint scores.

4.12.1 Our Matching Process

The process of matching is composed of two steps:

1. **Applying the Correlation Matching:**
   In this case, every point in the set A (which represents the first image) is connected to one or more points in the set B (second image) with the constraint that the correlation score is above or under the set threshold, depending on the correlation measure used.

2. **Applying a Matching Constraint:**
   When applying a matching constraint, the similarity measure score based on the applied constraint is associated with every edge and only edges consistent with the constraint will survive. In other words, only edges which have similarity measure score above or under the set threshold will survive.

Technically, this graph matching algorithm is implemented as an ***adjacency list*** [2]. The optimization problem, while implementing constraint algorithms, is not a concern for us. All algorithms are adapted from existing literature and customized to our case. The programming language used to implement the algorithms is *C++ Builder* on *Windows95* platform. *C++ Builder* provides a visual environment for development, figure 4.9 shows a sample view of our Application Interface.

[2] An adjacency list is a list of points where each point has a list of candidates.

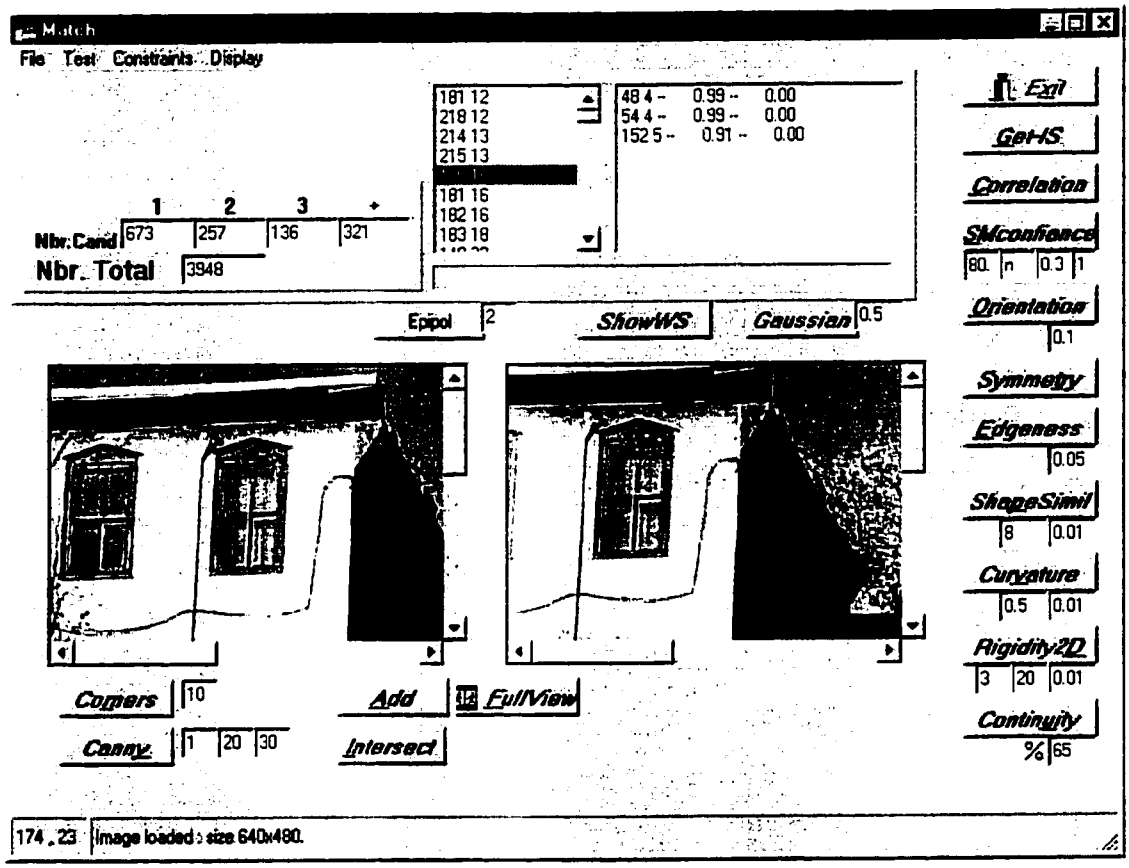


Figure 4.9: A sample view of our application interface

4.12.2 A Typical Session

The idea is to design correspondences between two images using a correlation technique and to apply a matching constraint on these correspondences.
In a typical session on how to use our application, the steps a user can perform are listed in the following:

1. *Opening* an image: in the menu, you choose *File-Open* to open each image. You are then asked, for each image, to choose where to load this image (on left panel or on right panel).

2. Noise suppression: you can apply a gaussian filter, by clicking on the *Gaussian* button, to suppress noise if necessary.

3. *Cornering* images: you enter the *cornering* threshold and click on the *Corners* button. Then you press on the *Add* button to capture the selected corner points if pleased with them. Otherwise you redo the *cornering*.

4. *Cannying* images: you enter the required parameters and click on the *Canny* button. Then you press on the *Intersect* button to capture corner points that are also edge points. Here, also, you can redo the *cannying*.

5. Points of interest: now that you like the interest points you load them in the *InterestSet* by clicking on the button *Get-IS*.

6. Correlation: to apply the correlation process press the button *Correlation* and a *DialogBox* pops up. You select which correlation to apply, VNC or SSD. You enter $(dminX, dmaxX, dminY, dmaxY)$ [3] to define the search window, (m, n) to define the correlation window and finally the correlation threshold. When the correlation is done you can click on the *ShowWS* button to display the list

[3] see section4.3

of points and their candidates. You also get the total number of matches with 1, 2, 3 and more candidates and the overall total number of matches.

7. Constraints: at this stage, you just decide which matching constraint to apply, you enter the required parameters and you click on the designated button.

By clicking on the *Test* menu, you can also apply a function to generate the necessary files which are used to plot the graphs. The graphs are generated by using "*MATLAB 5*" on a Unix workstation.

4.13 Summary

In the first section, the matching process is introduced. This section explains in detail how the framework (1.2), presented in the first chapter, works. We also presented detailed algorithms of the existing constraints in the literature which we selected to study: (sometimes with a slight variation), such as: correlation, edgeness, figural continuity and 2D rigidity; or with a whole new approach based on some existing ideas: shape similarity. Finally, the last section provides some information on the implementation details.

Chapter 5

Image Test Set and Feature Points Validation

5.1 Introduction

In this chapter, the set of images used for testing will be presented at a first stage in the first section. At a second stage, some information about the calibration process used to obtain the camera geometry will also be reviewed. Finally in the second section, we will introduce our validation process that allows us to measure the performance of each constraint.

5.2 The Image Test Set

In order to investigate the performance of each constraint 6 black and white pairs of images were used. They are shown in Figure 5.1. The last two look similar, but the purpose of using this set of pairs is to investigate the effect of lighting variation.

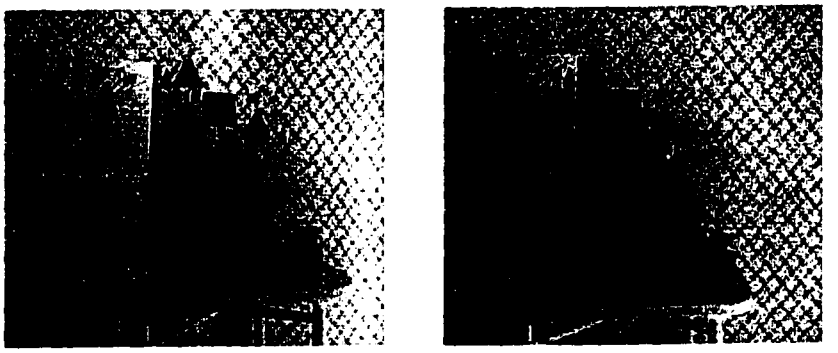
(a) STAIRS

(b) HOUSE

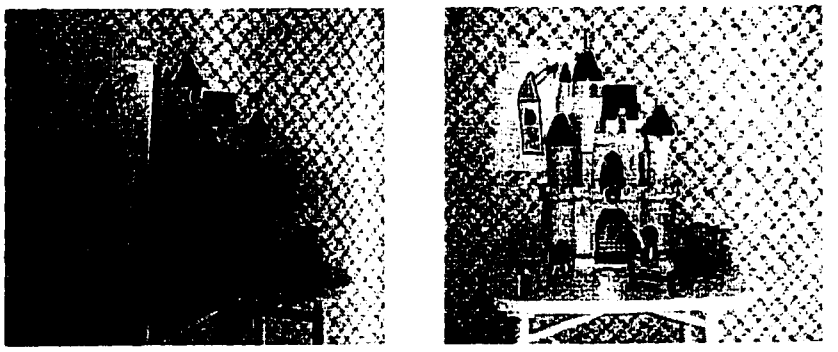
(c) VAN

Figure 5.1: Pairs of images used in testing.

(c) TANK



(d) TOYS Dark



(e) TOYS

Figure 5.1: Pairs of images used in testing (con't)

	Maximum	Average	Minimum
STAIRS	357.136	203.598	45.45
HOUSE	558.79	278.76	30.805
VAN	572.37	313.62	45.31
TANK	366.28	215.28	7.21
TOYS Dark	552.42	339.53	35.85
TOYS	457.45	215.23	26.17

Table 5.1: Maximum, average and minimum disparity of all the pairs of images.

5.2.1 Source of Images

Our sample pairs of images were taken using a digital color camera MVC-FD71[1] (Digital MAVICA) by Sony and saved in *jpg* format and 640 × 480 size. The images were then converted into black and white *pgm* format using "xv 2.21 by John Bradle" on a Unix workstation.

Most of the images originated in the city of Ottawa, the national capital of Canada. The STAIRS scene represents the stairs that lead to "*Colonel By*" Hall, a building of the campus of the university of Ottawa from "*King Edward Avenue*". The HOUSE and VAN scenes were taken on "*Templeton* Street", a nearby street. The TOYS pairs are baby's room scene. These scenes were captured by performing a translation displacement of the same digital camera. For instance, the TANK scene was taken in front of the "*Manége militaire de Hull*" Hull-Québec, but a rotation displacement of the camera was involved.

The disparity of the images is shown in the table 5.1: Three parameters are presented in this table: maximum, average and minimum disparities of a set of matched points uniformly spread on the pairs of images. The disparity at a match was computed as

[1]The focal distance of the camera is 80 mm.

the magnitude, in pixels, of the Euclidean distance between the points that form a match.

5.2.2 Images Calibration

The 6 image pairs have been calibrated using the TotalCalib software[2]. We therefore obtain for each of these images the value of the **F** matrix(see Chapter 3 about camera geometry). The following table shows the computed fundamental matrices associated to each image pair.

STAIRS			HOUSE		
0.000000449757	-0.00000593013	-0.00234925	-0.000000602832	-0.0000554185	0.00342329
0.0000128299	0.000000318709	-0.0653077	0.0000591001	-0.000000466277	0.062313
-0.0000693455	0.0637986	0.995821	-0.00365208	-0.0661486	0.99585
VAN			**TANK**		
-0.000000183828	0.00000808071	-0.002269	0.0000000932258	0.0000398497	-0.0122786
0.00000321954	-0.000000065046	0.0396161	-0.00000767576	0.00000463382	-0.454352
0.0000780706	-0.0432039	0.998278	0.00335904	0.44208	0.773283
TOYS-DARK			**TOYS**		
0.0000133989	-0.00000561507	-0.0457944	-0.0000168128	-0.00002.59346	0.0676743
0.00005.5816	-0.0000181285	-0.188028	-0.000051502	0.0000180586	0.294481
0.0290769	0.180637	0.963885	-0.0562643	-0.339099	0.889127

Table 5.2: Fundamental matrices associated to each image pair.

In this step, point matches have been selected manually, the accuracy of the parameters should be good. Recall that this information will not be used in the matching procedures studied in this work. It is rather used to validate the accuracy of the resulting match set after applying a match constraint.

[2]Totalcalib can be downloaded from the following site:
"http://www-sop.inria.fr/ robotvis/personnel/sbougnou/TotalCalib/"

Four methods were suggested to compute the matrix **F**: SVD, Min Dists, LMeds and M Estimator. We choose the best approximation of **F** according to the residual errors computed by the selected calibration method. They represent the means, over all the matched points, of the distances of a point to its corresponding epipolar line. To ensure this choice, we also use our observation of the visual result showing epipolar lines with the correspondence points, together, on the images. More than that, a refinement of this calibration is possible, using the resulting camera geometry to find new matches in the pair of images and recomputing **F**.

5.3 Validating Point Correspondences

One way of testing if a match $(\mathbf{m}, \mathbf{m}')$ is exact can be done by comparing the intensity gray level at the points forming the match: $\|I(\mathbf{m}) - I(\mathbf{m}')\|$. But this method is weak, since even at the same lighting conditions there could be a high intensity variation from one image to the other.

The approach we suggest here to validate the resulting match set is based on the inter-image geometry namely the fundamental matrix **F**. Equation (3.11) leads us to two obvious observations that will be the basis of our validation procedure:

- when two points form an exact match, then one should lies on the epipolar line of the other. In practice, because of the unavoidable inaccuracy, it will be very close from it.

- when two points are wrongly matched, and if the matching process has been done without any knowledge of the camera geometry, then most often, the corresponding points will fall sufficiently far from their respective epipolar lines to be detected as inexact. It is understood that it may happen that, by chance, some false matches will lie very close to their corresponding epipolar line, but

we believe that this fortuitous event will be sufficiently unlikely to be neglected.

Consequently, we propose and apply the following scheme to evaluate the validity of a matching constraint.

1. Starting from a first set of candidate point matches, (obtained using an unconstrained correlation, see section 4.3), the constraint under study is applied.

2. The result is a new set of point matches, all consistent with the constraint.

3. In order to evaluate the ability of this constraint to increase the quality of a match set, we estimate the number of false matches before and after the application of the constraint. Comparing these numbers gives us an objective measure of the constraint relevance.

4. To estimate the number of false matches in a given set, we proceed as follows:

   i. for each matched pair, compute the epipolar line associated with one of the points (using the $\mathbf{F}$ matrix as obtained by calibration, [Luo93])

   ii. compute the distance between the other point of the pair and the computed epipolar line. This distance is given by the following formula [Zha94]:

$$d(\mathbf{m}', \mathbf{Fm}) = \frac{\left|\mathbf{m}'^T \mathbf{Fm}\right|}{\sqrt{(\mathbf{Fm})_1^2 + (\mathbf{Fm})_2^2}} \tag{5.1}$$

   where $(\mathbf{m}, \mathbf{m}')$ forms the match pair, $(\mathbf{Fm})_1$ and $(\mathbf{Fm})_2$ represent the first and second element of the vector $\mathbf{Fm}$ respectively.

   iii. if this distance is less than 3 or 2 pixels then the match is considered as valid.

5.4 Summary

We presented in this chapter the image test set in the first part and the calibration software used to compute **F** for validation purposes. The second section explains the algorithm to follow in order to determine, after applying one of the constraints, whether or not a match pair is relevant using the defined distance in (equation 5.1).

Chapter 6

Summary of the Results

6.1 Introduction

This chapter shows the experimental results we have obtained by applying the two correlations VNC and SSD on all the image pairs presented in chapter 5 (STAIRS, HOUSE, VAN, TANK, TOYS-DARK and TOYS). It also shows the performance of each of our constraints. The goal was to test and validate each constraint presented in chapter 4 with respect to their ability to discard false matches while keeping a substantial amount of good matches.

6.2 Definitions

To ease the understanding, definitions of some keywords used in this chapter are given here:

Candidate point:	is a point in the second image that corresponds to a point in the first image.
Candidate match:	a point from an image with all its corresponding points in the second image form a candidate match.
Ambiguous match:	is a candidate match where there exist more than one corresponding point.
Non-ambiguous match:	is a candidate match where a point has one and only one corresponding point.
False match:	is a matched pair from the $WorkSet$ that does not satisfy our validation constraint.

6.3 Correlation Results

The first two tests show the results when the two correlation functions, VNC (equation 4.1) and SSD (equation 4.4), are applied alone using four different threshold values. For each image pair, three graphs are produced (see figures A.1 and A.2). The graphs are included in the appendix A.

- The first graph shows the total number of feature points for which at least one candidate point has been found in the other image (*dotted line*). It also shows the number of feature points having exactly one candidate in the other image i.e. the number of *non-ambiguous matches* (*solid line*).

- In order to estimate the number of false matches present in a given $WorkSet$, we count the number of matching pairs in this $WorkSet$ for which the distance between the candidate point and the corresponding epipolar line is greater than two pixels (equation 5.1). The graph shows the total number of false matches over all matched pairs (*dotted line*), the number of false matches for *non-ambiguous* ones (*solid line with o sign*) and finally for *ambiguous matches* only the number

of false matches with maximum correlation score is shown (*solid line with + sign*).

- The last graph shows the mean value of the distances between all the matches and the epipolar line on which they should lie (*solid line*). The median distance and the standard deviation are also shown.

The same three graphs will be presented for all matching constraints experiments.

It is interesting to note that consistent results are obtained with both correlations as they seem to lead to similar performances. This is expected when lighting conditions are similar from one image to the other (which is the case for most of the test images). For the last pair shown in Figure 5.1(e) the lighting variation prooved to weaken the performance of SSD.

Although it is computationally more expensive, the VNC correlation has the advantage that the adequate threshold value that should be used for a particular image is more predictable. In the rest of this study, we will use the $WorkSet$ of candidate matched points obtained when applying the VNC correlation with a threshold value of 0.85. For SSD the threshold will vary between 20000 and 30000 depending on the pair of images. This gives us initial $WorkSets$ for both correlations with the following characteristics shown in table 6.1.

Applying VNC

	Stairs	House	Van	Tank	Toys Dark	Toys
Total number of candidate matches with at least one candidate	370	932	735	987	572	552
Total number of non-ambiguous matches	142	525	265	381	197	144
Estimated number of false matches	1742	1138	2318	1683	1356	2655
Estimated number of false matches (non-ambiguous matches only)	45	67	128	117	124	113

Applying SSD

	Stairs	House	Van	Tank	Toys Dark	Toys
Total number of candidate matches with at least one candidate	474	431	238	998	546	863
Total number of non-ambiguous matches	224	348	120	406	135	362
Estimated number of false matches	1558	181	587	1275	2815	2226
Estimated number of false matches (non-ambiguous matches only)	43	33	47	108	66	352

Table 6.1: Initial *WorkSets* characteristics.

6.4 Matching Constraints Results

From our experiments, we conclude that the confidence measure (see figure A.3) seems to be an effective way to reduce the proportion of ambiguous matches (points with more than one candidate) in a set of candidate matches. Also, it is the only constraint in our test that has been able to reduce (in some cases) the number of false matches to zero. However, the price to pay for this is an important reduction in the total number of matched pairs.

Table 6.2 and the rest of the tables in this section show the *WorkSets* characteristics before (within parenthesis) and after applying a given constraint when fixing the threshold value to what we consider to be an adequate value. In these tables we show only the image where the constraint seems to be the less effective[1], the most effective

[1] refer to section 6.5.1 for the definition of effectiveness

and the average over all images.

Applied with VNC

	Stairs	Van	Average
Total number of candidate matches with at least one candidate	207(370)	253(735)	356(691)
Total number of non-ambiguous matches	78(142)	120(265)	176(276)
Estimated number of false matches	1340(1742)	66(2318)	599(1815)
Estimated number of false matches (non-ambiguous matches only)	4(45)	8(128)	18(99)

Table 6.2: $WorkSets$ characteristics before and after applying the confidence measure. Threshold is set to $th = 2$

Orientation (figure A.4) and edgeness (figure A.6) seem to be less effective in terms of false match reduction. One reason for that might be that these two constraints are, more or less, implicitly taken into account by the correlation process. Indeed, in order to produce correlation values close to unity (in the case of VNC), the edgeness and orientation of the considered points must be similar. However, the use of these constraints appears to be justified if one wants to disambiguate matches with several candidates. Shape similarity (figure A.5) looks to have the same behavior as the orientation, but performs better over all the images, especially on the STAIRS image which we consider as an ambiguous image because of its texture.

In the case of edgeness false matches curves (figure A.6) for the HOUSE image drop better and the distance mean is the best compared to the other images whereas it is in its worst case for the TOYS image. The same observation holds for the orientation and shape similarity, but it does not hold in the case where the shape similarity is applied with VNC correlation. It seems to be more effective on STAIRS image instead. In the following tables we present the resulting workSet characteristics for all three constraints.

Applied with VNC

	Toys	House	Average
Total number of candidate matches with at least one candidate	360(552)	724(932)	529(691)
Total number of non-ambiguous matches	147(144)	487(525)	264(276)
Estimated number of false matches	1149(2655)	482(1138)	881(1815)
Estimated number of false matches (non-ambiguous matches only)	121(113)	56(67)	85(99)

Applied with SSD

	Toys	House	Average
Total number of candidate matches with at least one candidate	787(863)	383(431)	534(592)
Total number of non-ambiguous matches	337(362)	316(348)	248(266)
Estimated number of false matches	1938(2226)	141(181)	1248(1440)
Estimated number of false matches (non-ambiguous matches only)	325(352)	26(33)	104(109)

Table 6.3: *WorkSets* characteristics before and after applying the edgeness constraint. Threshold is set to $th = 0.02$

Applied with VNC

	Toys	House	Average
Total number of candidate matches with at least one candidate	534(552)	848(932)	650(691)
Total number of non-ambiguous matches	161(144)	498(525)	279(276)
Estimated number of false matches	2299(2655)	913(1138)	1600(1815)
Estimated number of false matches (non-ambiguous matches only)	123(113)	68(67)	101(99)

Applied with SSD

	Toys	House	Average
Total number of candidate matches with at least one candidate	572(863)	392(431)	509(592)
Total number of non-ambiguous matches	329(362)	323(348)	263(267)
Estimated number of false matches	1033(2226)	128(181)	998(1440)
Estimated number of false matches (non-ambiguous matches only)	316(352)	33(33)	106(108)

Table 6.4: *WorkSets* characteristics before and after applying the orientation constraint. Threshold is set to $th = 0.31$

Applied with VNC

	Toys	Stairs	Average
Total number of candidate matches with at least one candidate	406(552)	318(370)	562(691)
Total number of non-ambiguous matches	131(144)	160(142)	259(276)
Estimated number of false matches	1250(2655)	474(1742)	905(1815)
Estimated number of false matches (non-ambiguous matches only)	98(113)	36(45)	81(99)

Applied with SSD

	Toys	House	Average
Total number of candidate matches with at least one candidate	514(863)	399(431)	471(592)
Total number of non-ambiguous matches	256(362)	332(348)	238(267)
Estimated number of false matches	1111(2226)	115(181)	802(1440)
Estimated number of false matches (non-ambiguous matches only)	250(352)	24(33)	83(108)

Table 6.5: *WorkSets* characteristics before and after applying the shape similarity constraint. Threshold is set to $th = 10$

With the curvature constraint (figure A.7), some reductions in the number of false matches are obtained. However, as expected, due to the difficulty to compute reliable estimates of curvature, which depends on the gradient and second derivative of the image, some instabilities have been observed. Again the HOUSE image seems to be the most effective and the less effective one is the TOYS image. But for SSD correlation it is the VAN image that is more effective.

Applied with VNC

	Toys	House	Average
Total number of candidate matches with at least one candidate	168(552)	176(932)	224(691)
Total number of non-ambiguous matches	110(144)	151(525)	117(276)
Estimated number of false matches	211(2655)	106(1138)	287(1815)
Estimated number of false matches (non-ambiguous matches only)	92(113)	61(67)	72(99)

Applied with SSD

	Toys	Van	Average
Total number of candidate matches with at least one candidate	124(863)	86(238)	163(592)
Total number of non-ambiguous matches	114(362)	53(120)	115(267)
Estimated number of false matches	130(2226)	89(587)	176(1440)
Estimated number of false matches (non-ambiguous matches only)	110(352)	25(47)	67(108)

Table 6.6: *WorkSets* characteristics before and after applying the curvature constraint. Threshold is set to $th = 0.1$

As implemented, the 2D rigidity constraint (Figure A.8) appears to be more severe. It indeed drastically reduces the number of matched pairs without leading to any significant false matche removal. A rigidity 3D constraint that takes into account the perspective distortion would probably be more efficient [Hu94].

Applied with VNC

	Toys Dark	House	Average
Total number of candidate matches with at least one candidate	135(572)	227(932)	194(691)
Total number of non-ambiguous matches	87(197)	165(525)	117(276)
Estimated number of false matches	170(1356)	99(1138)	215(1815)
Estimated number of false matches (non-ambiguous matches only)	58(124)	22(67)	51(99)

Applied with SSD

	Toys	House	Average
Total number of candidate matches with at least one candidate	196(863)	247(431)	207(592)
Total number of non-ambiguous matches	133(362)	193(348)	131(267)
Estimated number of false matches	288(2226)	73(181)	208(1440)
Estimated number of false matches (non-ambiguous matches only)	131(352)	14(33)	63(108)

Table 6.7: *WorkSets* characteristics before and after applying the 2D rigidity constraint. Threshold is set to $th = 20$

Finally, the figural continuity constraint (Figure A.9) eliminates several false matches but not enough, even with a severe threshold. This is mainly due to the fact that when detecting edges, broken contours are often obtained. Therefore, it is not unusual to find points located on the same object contour that are in fact lying on different pieces of the contour. These points are mostly rejected by the constraint.

Applied with VNC

	Toys	House	Average
Total number of candidate matches with at least one candidate	524(552)	809(932)	603(691)
Total number of non-ambiguous matches	149(144)	476(525)	258(276)
Estimated number of false matches	2332(2655)	868(1138)	1479(1815)
Estimated number of false matches (non-ambiguous matches only)	113(113)	46(67)	85(99)

Applied with SSD

	Toys	House	Average
Total number of candidate matches with at least one candidate	858(863)	378(431)	537(592)
Total number of non-ambiguous matches	359(362)	310(348)	248(267)
Estimated number of false matches	2221(2226)	151(181)	1345(1440)
Estimated number of false matches (non-ambiguous matches only)	349(352)	24(33)	102(108)

Table 6.8: *WorkSets* characteristics before and after applying the figural continuity constraint. Threshold is set to $th = 75\%$

6.5 Comparison Criteria

From the graphs shown in the appendix it is obvious to notice that they differ from one constraint to the other for the same image pair and from one image pair to the other for the same constraint.

When comparing the different constraints, the questions that come to the mind are: how many false matches are still present in the $WorkSet$, and how many candidate matches are eliminated in total for ambiguous or non-ambiguous matches. Three criteria must be taken into consideration to make a comparison between the constraints: effectiveness, severity and also stability.

6.5.1 Effectiveness

The *effectiveness* of a constraint corresponds to the efficiency with respect to false match reduction as observed in the middle graphs. Most constraints perform adequately with this respect, notably the confidence measure, orientation, shape similarity, edgeness and curvature constraints that produce a significant reduction of this number (close to zero in some cases). The poor performance of the figural continuity must also be noticed. Although this is a potentially very powerful algorithm, its practical implementation is problematic because of the difficulty to obtain stable (and unbroken) curve detection.

6.5.2 Severity

The *severity* of a constraint is another important aspect. That is the number of candidate matches (good or not) which the constraint rejects as shown on the left graphs (note that there is clearly a tradeoff between criteria 1 and 2). In terms of severity, confidence measure performs well with a matched point reduction more or

less linear. Shape similarity and orientation also give good performances. Conversely, the 2D rigidity constraint becomes very severe below a certain threshold value.

6.5.3 Stability

Finally the *stability* of the results obtained by the application of a constraint must also be considered. Stability is related to the regularity of the different curves shown. A more stable constraint makes the choice of its associated threshold(s) easier to determine. Orientation, curvature, edgeness appear to be the most unstable constraints. This is mainly due to the fact that stable estimates of image derivatives are difficult to obtain.

6.6 Overall Evaluation

As noticed before both correlations seem to lead to the same results even when combined with a constraint. We then concentrate only on results when VNC is applied. As implemented, 2D rigidity is too severe and would not be useful in a matching algorithm. The same applies for the figural continuity because it showed less efficiency when combined with a constraint and we already know about its poor performance. In the case of curvature constraint and because of its instability, it is rather not a good idea to use it as implemented. It would be good if implemented with a more stable curvature function. Orientation, edgeness and shape similarity can play a good role that will be shown in a successful example in the next section. Definitely, the confidence measure is the best and its application is strongly suggested.

6.7 Combining Constraints

In the following we will show some results obtained by combining shape similarity, orientation and/or edgeness with the confidence measure. The same thresholds as before are used for confidence measure and shape similarity. Edgeness threshold is set to $th = 0.02$ and $th = 0.08$ for orientation.

Table 6.9 and 6.10 show the result for confidence measure, shape similarity and edgeness combined together and confidence measure, shape similarity and orientation respectively. The results are almost similar in terms of efficiency. In both cases a potential amount of false matches was eliminated.

	Stairs	House	Van	Tank	Toys Dark	Toys
Total number of candidate matches with at least one candidate	219(370)	646(932)	295(735)	596(987)	203(572)	120(552)
Total number of non-ambiguous matches	130(142)	467(525)	156(265)	316(381)	114(197)	61(144)
Estimated number of false matches	247(1742)	197(1138)	172(2318)	477(1683)	178(1356)	162(2655)
Estimated number of false matches (non-ambiguous matches only)	11(45)	27(67)	21(128)	52(117)	47(124)	36(113)

Table 6.9: *WorkSets* characteristics before and after applying the confidence measure, shape similarity and edgeness.

	Stairs	House	Van	Tank	Toys Dark	Toys
Total number of candidate matches with at least one candidate	139(370)	393(932)	230(735)	385(987)	175(572)	173(552)
Total number of non-ambiguous matches	84(142)	320(525)	138(265)	255(381)	124(197)	106(144)
Estimated number of false matches	157(1742)	166(1138)	145(2318)	277(1683)	116(1356)	222(2655)
Estimated number of false matches (non-ambiguous matches only)	13(45)	39(67)	24(128)	78(117)	50(124)	66(113)

Table 6.10: *WorkSets* characteristics before and after applying the confidence measure , shape similarity and orientation.

By combining all the above constraints and using the same thresholds we can get more reduction in false matches (see table 6.11).

	Stairs	House	Van	Tank	Toys Dark	Toys
Total number of candidate matches with at least one candidate	122(370)	344(932)	162(735)	301(987)	98(572)	67(552)
Total number of non-ambiguous matches	77(142)	294(525)	113(265)	217(381)	80(197)	43(144)
Estimated number of false matches	128(1742)	85(1138)	75(2318)	170(1683)	55(1356)	67(2655)
Estimated number of false matches (non-ambiguous matches only)	10(45)	31(67)	18(128)	60(117)	34(124)	26(113)

Table 6.11: *WorkSets* characteristics before and after applying the confidence measure, shape similarity, edgeness and orientation.

After an investigation on all pairs of images and applying all the above constraints together trying different kind of thresholds, the HOUSE image gave a very interesting result. The resulting match set of 35 matches is a *perfect match*. A *perfect match* is a set where all points in the first image are matched with one and only one corresponding point. More than that this corresponding points are validated as good matches.

	House
Total number of candidate matches with at least one candidate	35(932)
Total number of non-ambiguous matches	35(525)
Estimated number of false matches	0(1138)
Estimated number of false matches (non-ambiguous matches only)	0(67)

Table 6.12: *WorkSets* characteristics before and after applying the confidence measure, shape similarity $th = 5$, edgeness $th = 0.0001$ and orientation $th = 0.03$.

6.8 Summary

We presented in this chapter the experiments implemented on the matching constraints of chapter 4. The results were expressed in form of tables and three graphs for each constraint. The graphs are presented in the appendix and they show the behavior of those constraints by exposing the total number of candidate matches and non-ambiguous matches, the total number of false matches and the distance to the epipolar line. More graphs will be available in a technical report. Tables show the $WorkSets$ characteristics before and after applying one of the matching constraint or a combination of them. We also made a comparison between the constraints based on some criteria.

Chapter 7

Conclusion and Future Work

In this work, we have proposed a validation procedure that empirically evaluates the performance of a given matching constraint in the context of the feature-based correspondence problem. Analyzing every set of matched pairs, we were able to obtain an estimate of the number of false matches using the distance to the epipolar line as the decision criterion. Although this measure cannot be considered as exact, we believe that it is sufficient to validate the various match sets produced by the application of matching constraints. Obviously, this assumption remains valid as long as the first matching set has not been obtained using an epipolar constraint. In addition, the method also assumes that accurate estimates of the epipolar geometry parameters are available through a carefully performed calibration step.

We have tested a few constraints found in the literature on different pairs of images. Among which, the confidence measure and the shape similarity have been found to be the most effective. Orientation and edgeness are more or less effective but could be considered for a matching algorithm. Whereas, the curvature primarily depends on the image intensity derivatives. The figural continuity, when used with a stronger edge detector algorithm can be more efficient and can also be included by a matching algorithm. Unfortunately, the 2D rigidity is not sufficiently strong due to its severity

in eliminating false matches.

We have also investigated the effects of applying several matching constraints consecutively and we obtained interesting results. In general, they performed well in reducing false matches. For the HOUSE image, the results were very good and we obtained a set of 35 good matches. *Totalcalib* software suggests that to get a good camera calibration we need between 30 and 50 matches. In the case of the other pairs of images, we suspect that the poor performance obtained when combining matching constraints is due to texture in those images. For example, in the STAIRS image, the trees and the pavement produce significant ambiguity. The same problem occurred when similar patterns were repeated in the image. The TANK image has such a property where window patterns were similar. We believe that by exploiting some hierachical search methods where the matching is performed at different resolution levels, these two problems can be alleviated. However, our research already demonstrates the level of difficulty in the matching problem in computer vision.

Appendix A

Graphs

At each of the following pages, we will show a set of three graphs for the House image pair. The graphs of the other images and when the SSD correlation combined with a constraint is applied on the HOUSE pair will be available in a technical report untitled: "*Finding reliable point correspondences: an empirical evaluation of some matching constraint*". The curves are plotted at four different thresholds set on the correlation score when applied alone or set on the constraint when combined with the VNC correlation.

- The first graph shows:
  - the total number of feature points for which at least one candidate point has been found in the other image (*dotted line*)
  - the number of feature points having exactly one candidate in the other image i.e. the number of *non-ambiguous matches* (*solid line*).
- The second graph shows:
  - the total number of false matches over all matched pairs (*dotted line*)
  - the number of false matches for *non-ambiguous* ones (*solid line with o sign*)

  - the number of false matches with maximum correlation score for *ambiguous matches* (*solid line with + sign*)

- The last graph shows:

  - the mean value of the epipolar distances between all the candidate points and the epipolar line on which they should lie (*solid line*)
  - the median of epipolar distances (*dotted line*)
  - the standard deviation (*vertical lines*)

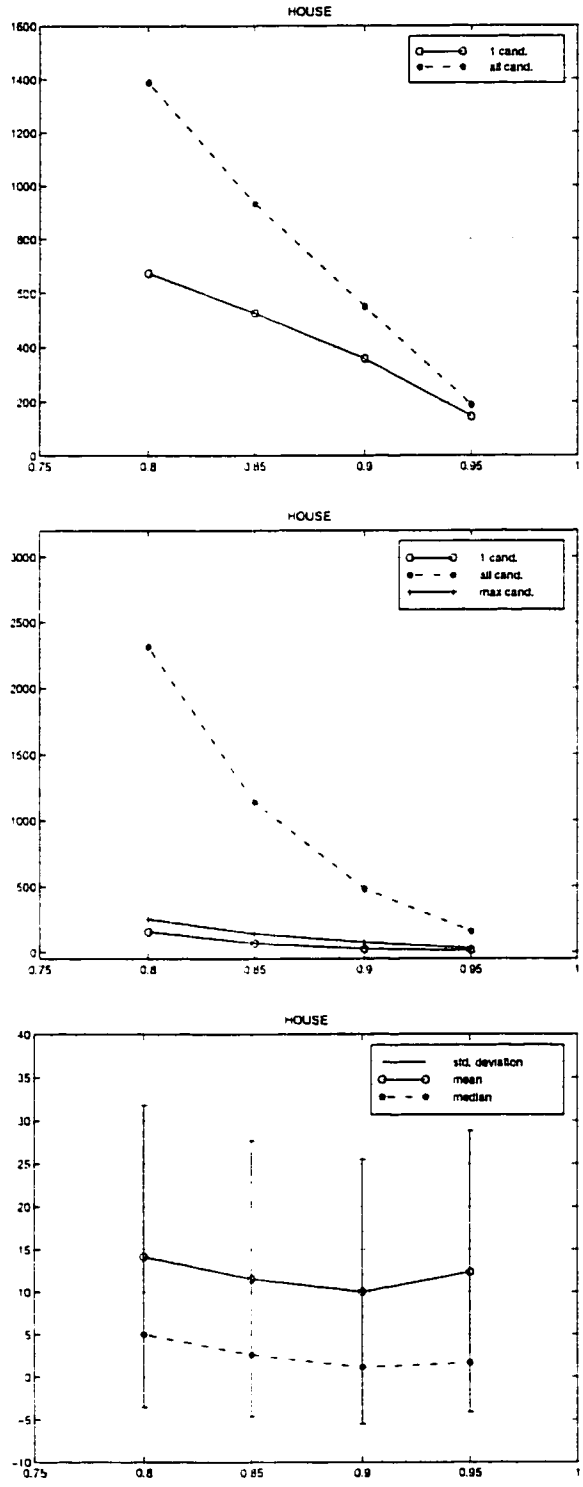


Figure A.1: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation.

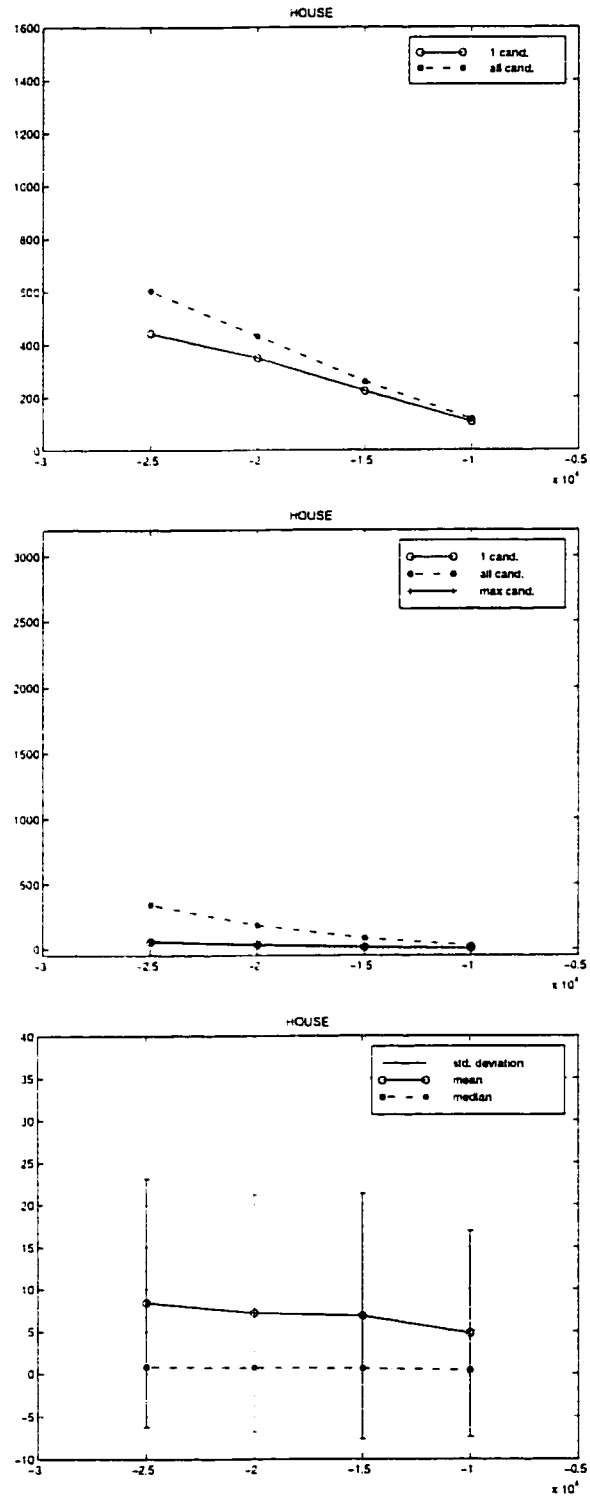


Figure A.2: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the SSD correlation.

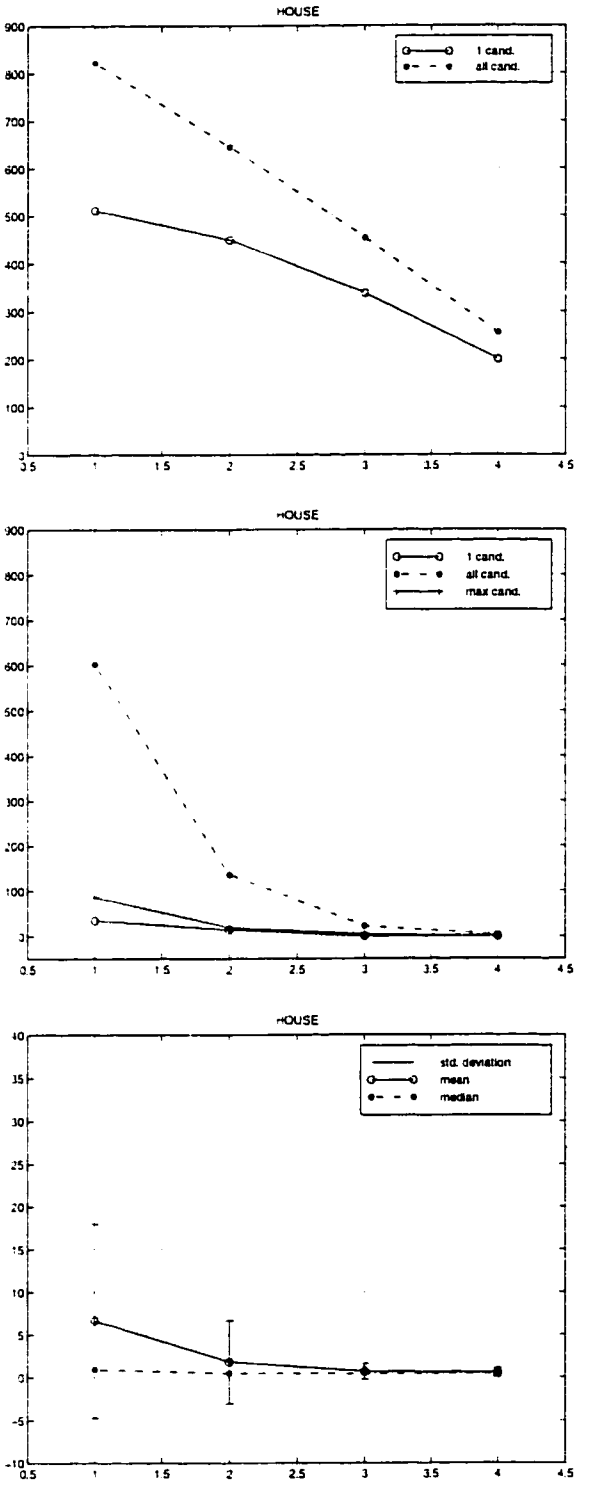


Figure A.3: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and confidence measure.

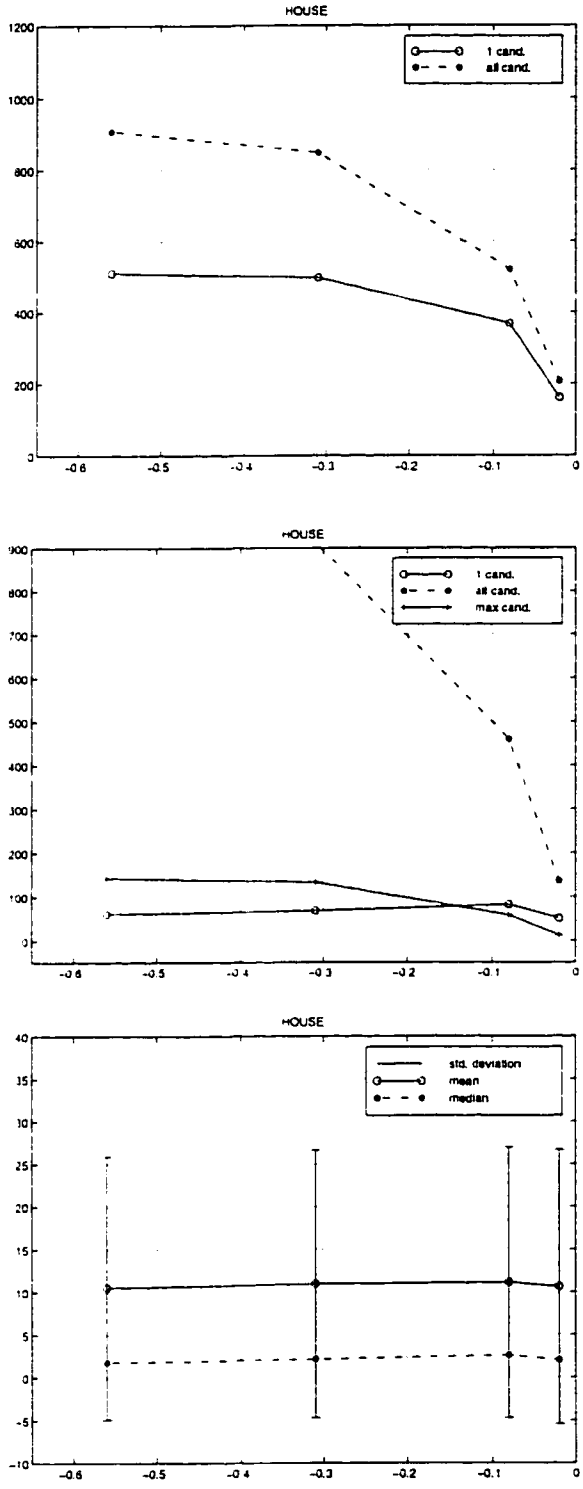


Figure A.4: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and orientation.

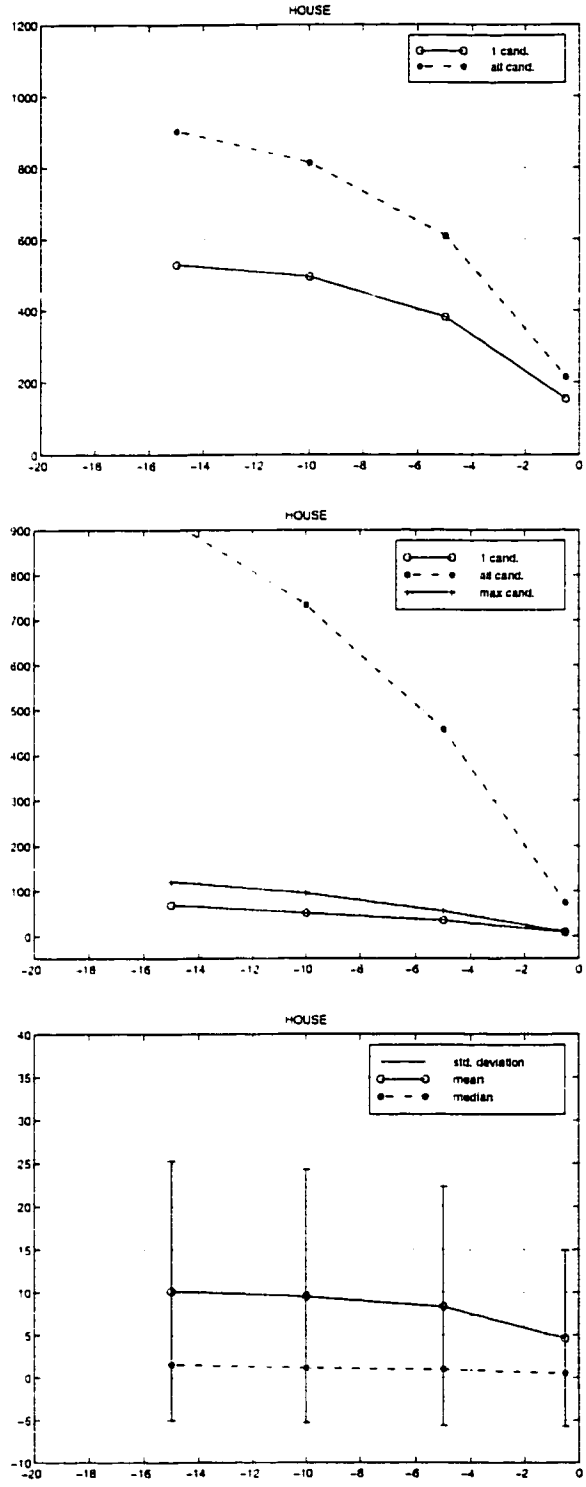


Figure A.5: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and shape similarity.

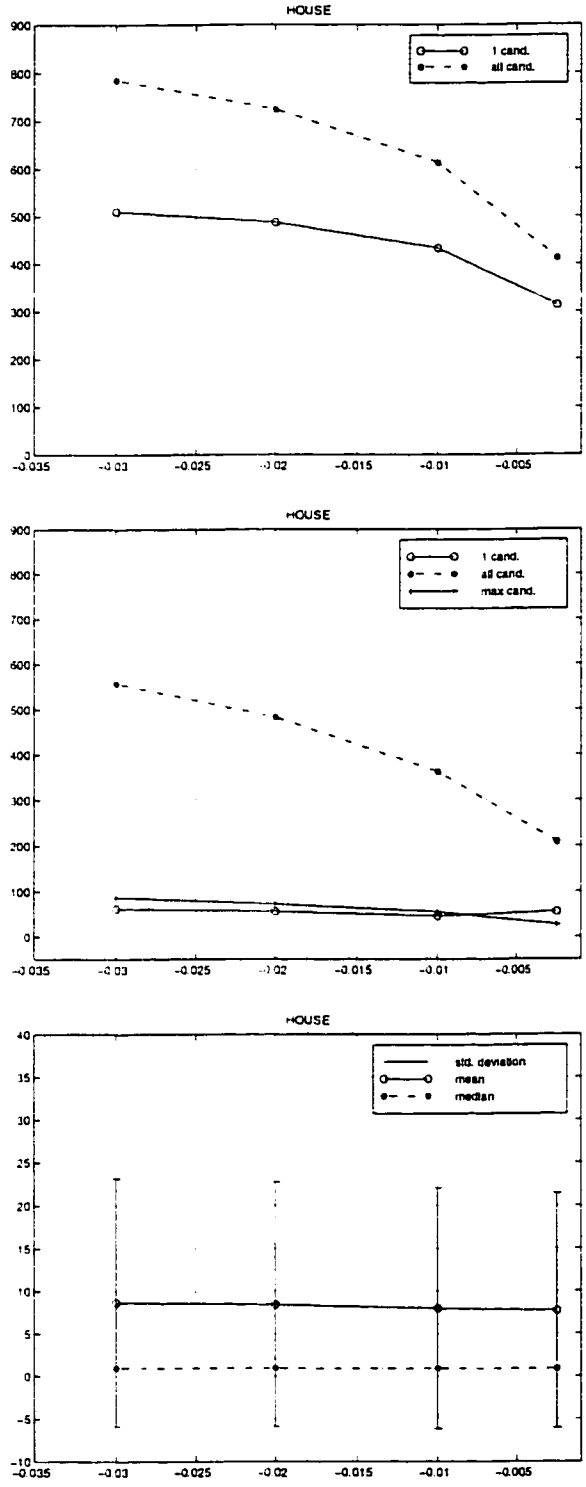


Figure A.6: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and edgeness.

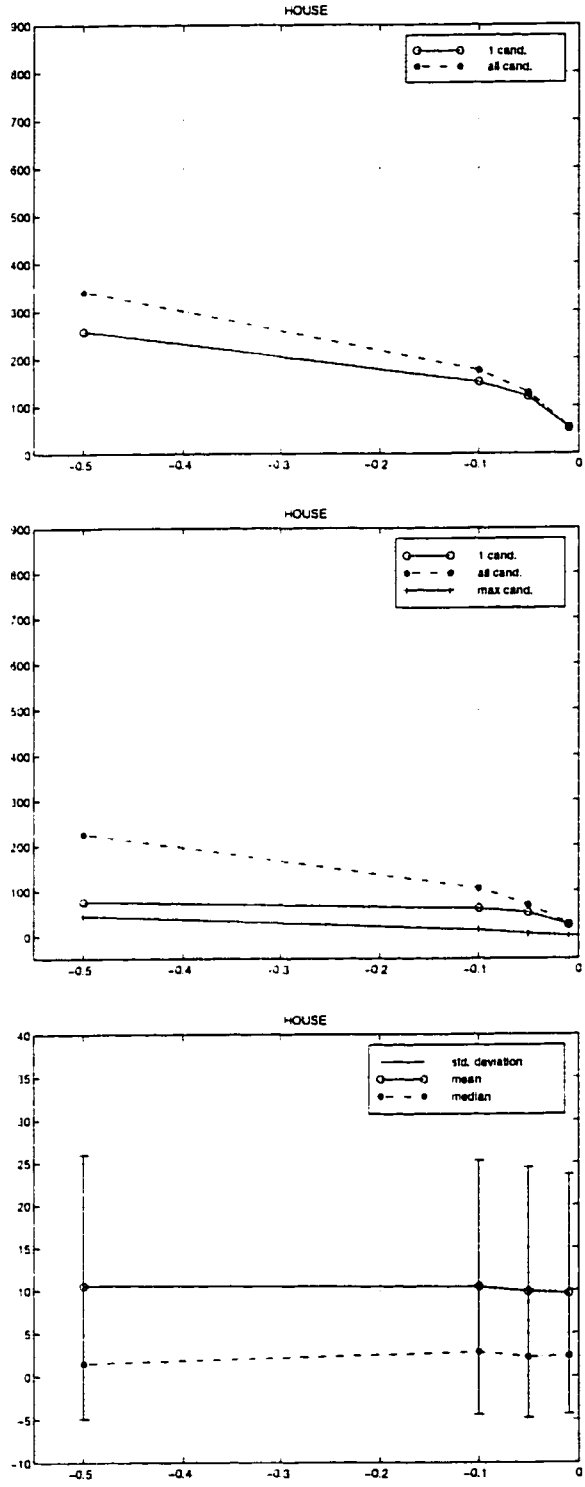


Figure A.7: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and curvature.

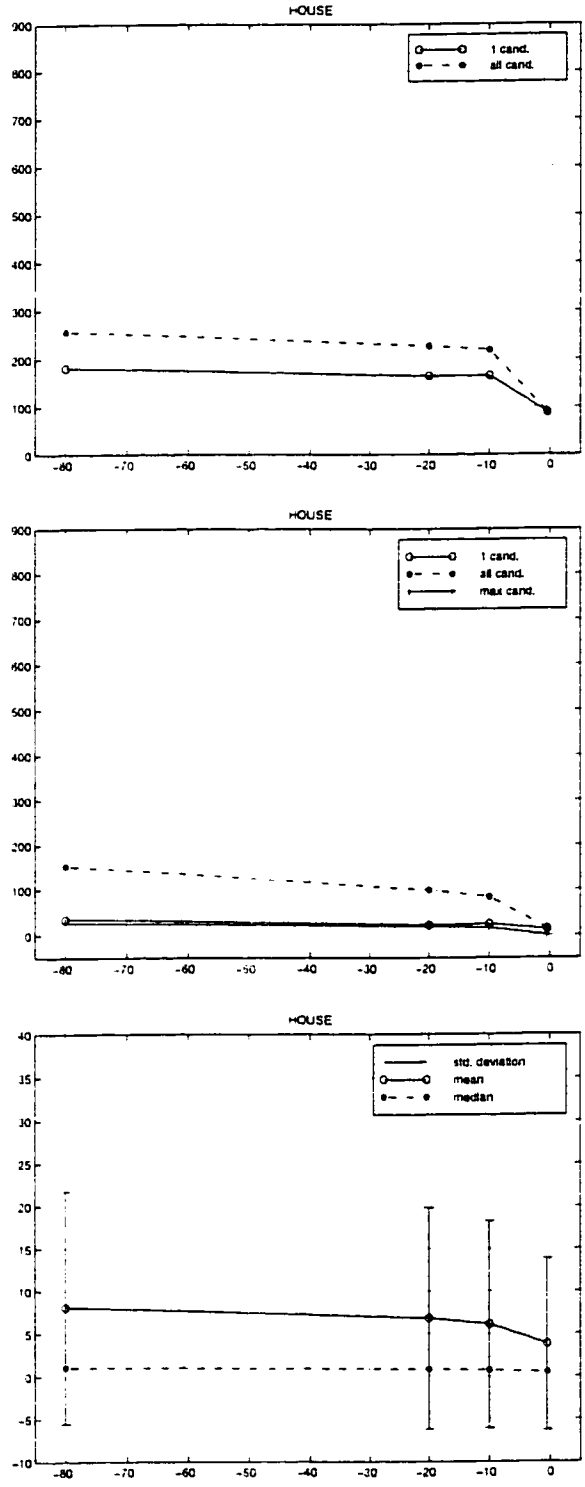


Figure A.8: These graphs show, for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and 2D rigidity.

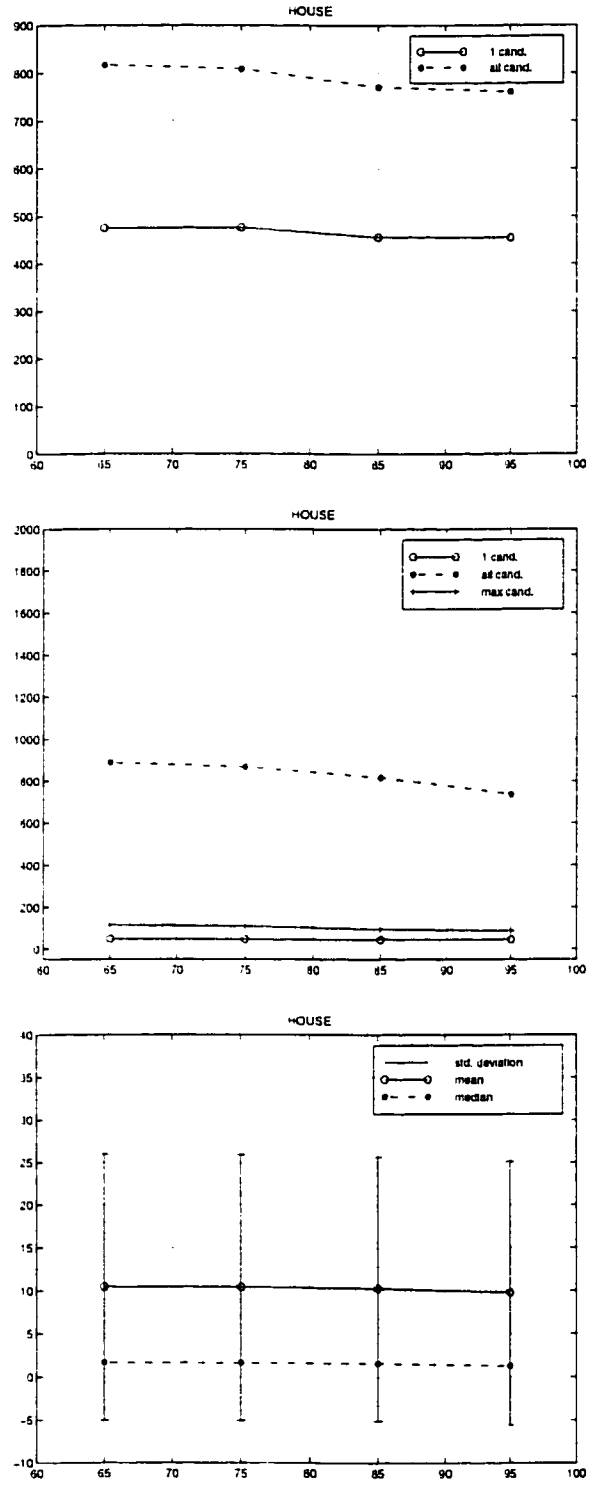


Figure A.9: These graphs show. for four different thresholds, the total number of matches remaining in a *WorkSet*, the total number of false matches and the epipolar distances when applying the VNC correlation and figural continuity.

Appendix B

Alternate graphs

We present here, in graph forms, an alternate way of showing the behavior of a constraint in eliminating false matches. Each figure shows:

- the total number of false matches in a *WorkSet* which is represented by the horizontal axis.
- the total number of matches in a *WorkSet* which is represented by the vertical axis .

The *solid line* presents the *HOUSE* pair and the *dotted line* presents the *TANK* pair. Note that each plotted point corresponds to a certain threshold on the applied constraint for which its value is also shown on the figure. In this case we do not have detailed information about the behavior of a constraint on a certain pair of images. Whereas in the previous graphs, more details are available in the graphs as for ambiguous matches versus non-ambiguous ones and also the availability of the epipolar distance information. However a nice curve or a good constraint should result in a curve that resembles in shape form to the one of confidence measure when applied on the HOUSE pair.

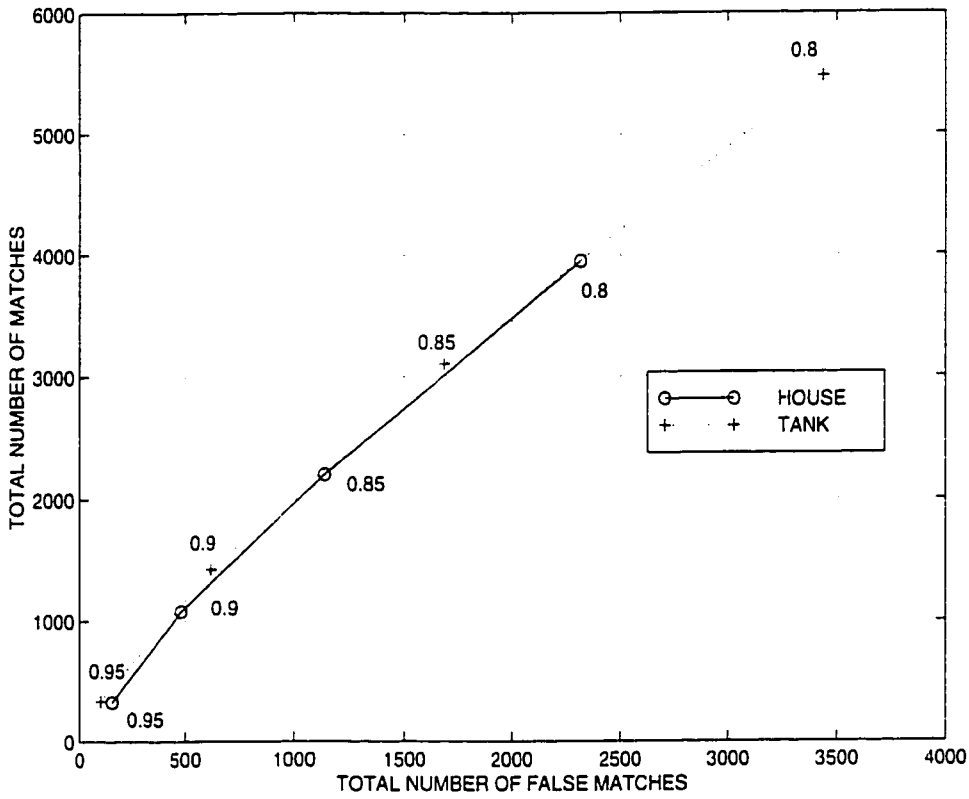


Figure B.1: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation.

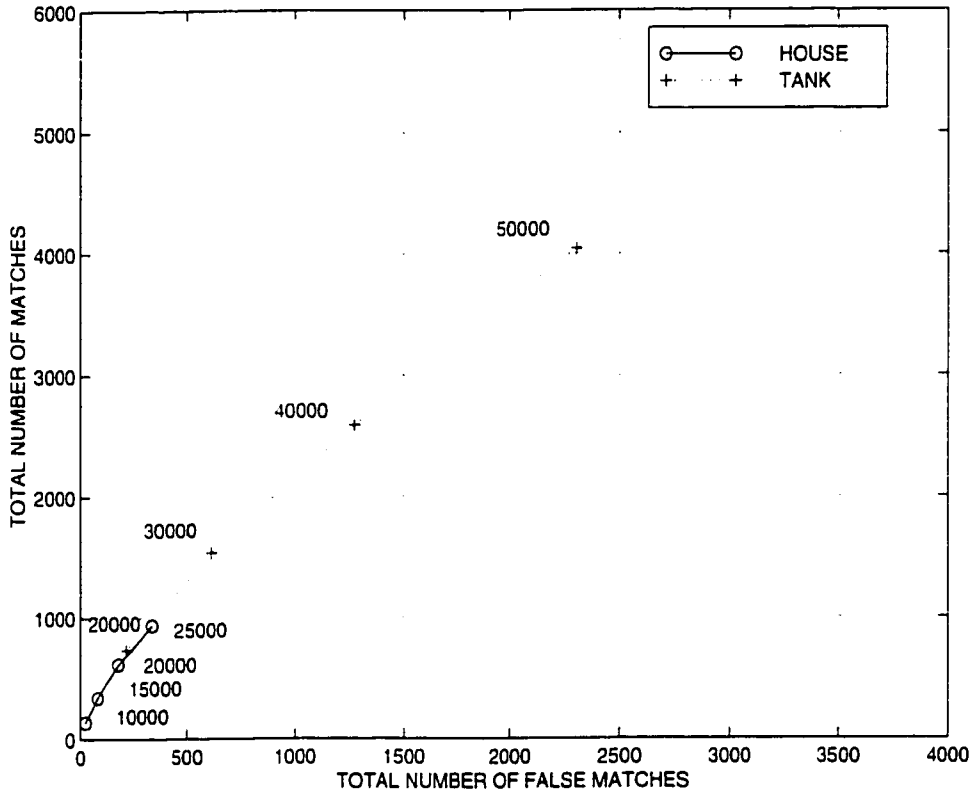


Figure B.2: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the SSD correlation.

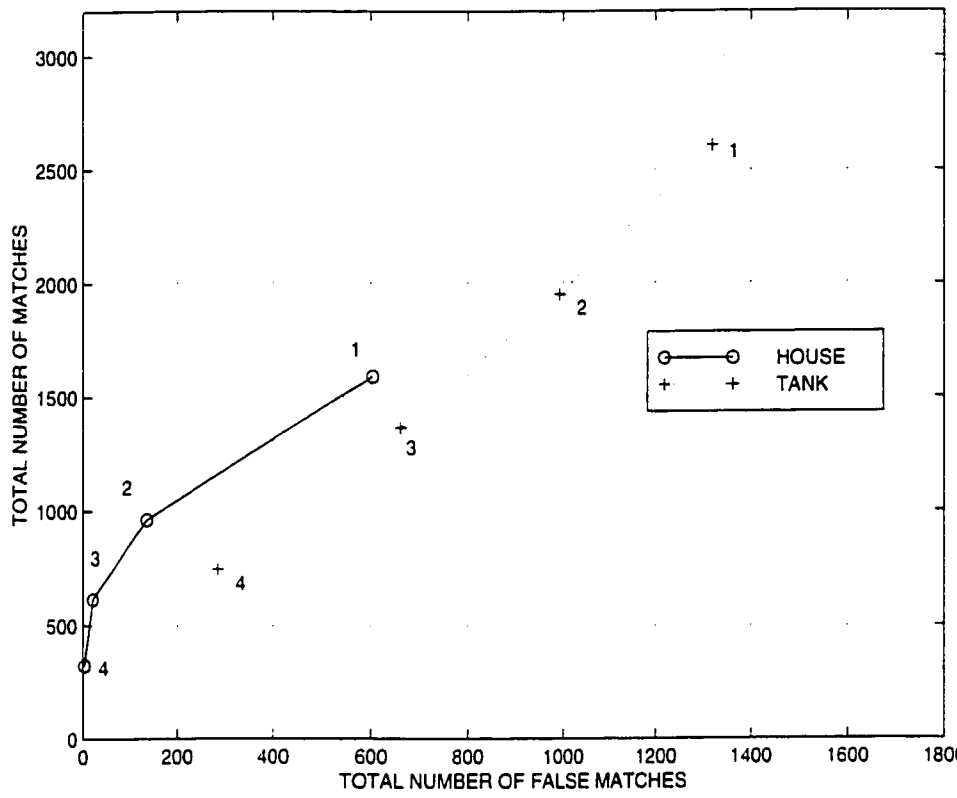


Figure B.3: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and confidence measure.

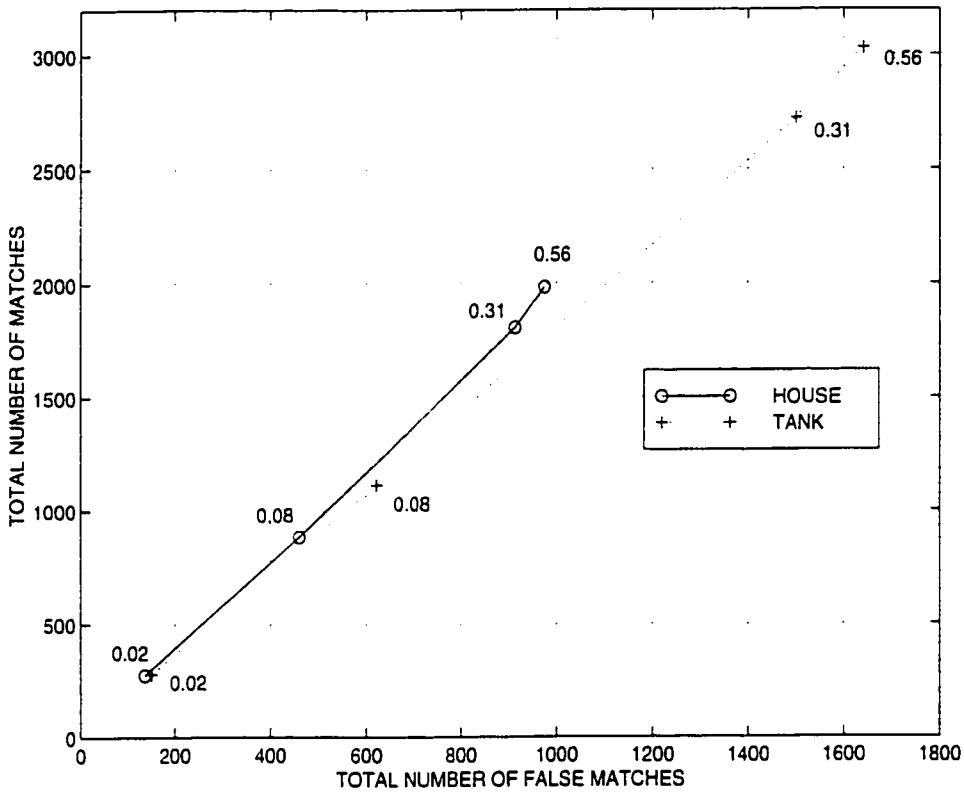


Figure B.4: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and orientation.

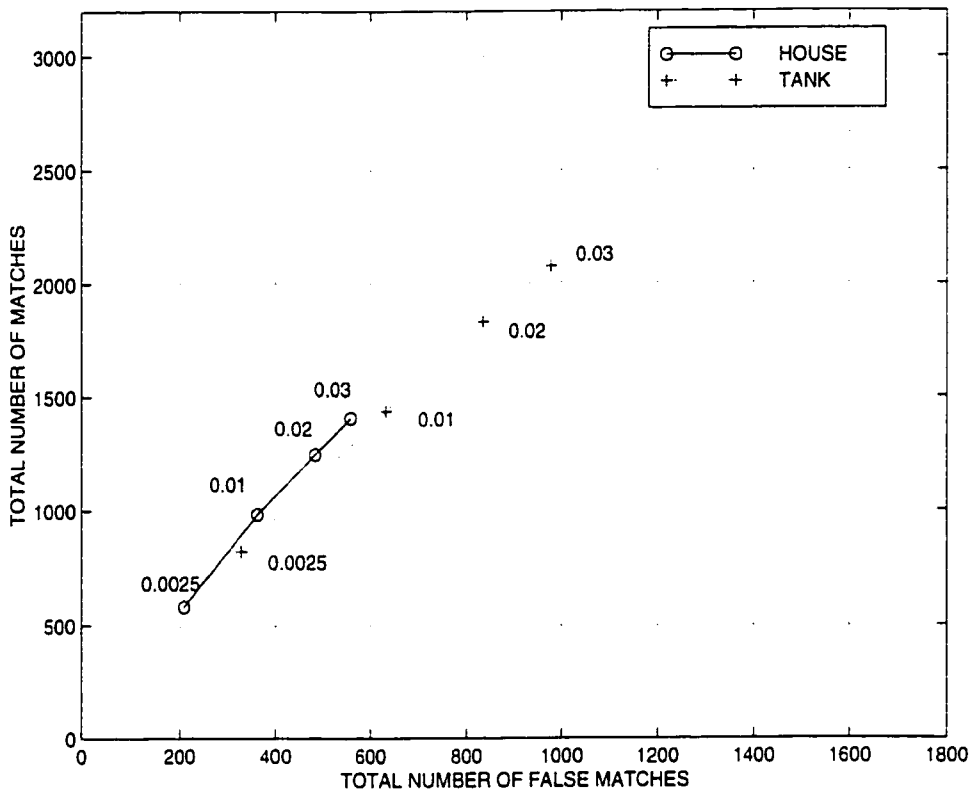


Figure B.5: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and edgeness.

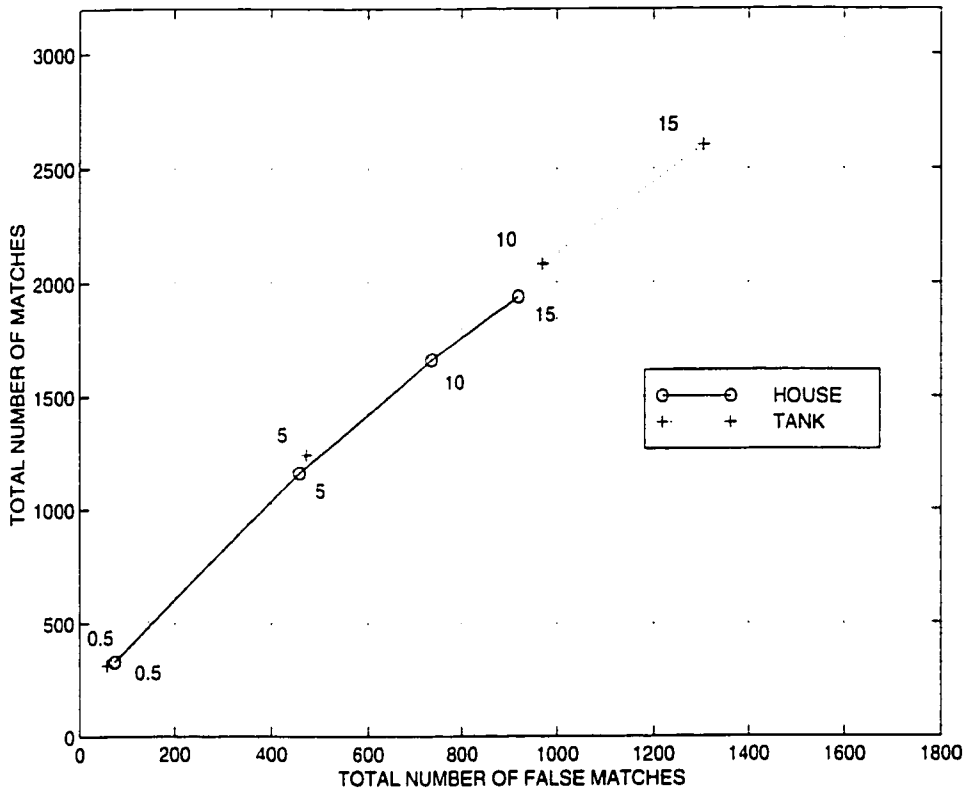


Figure B.6: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and Shape similarity.

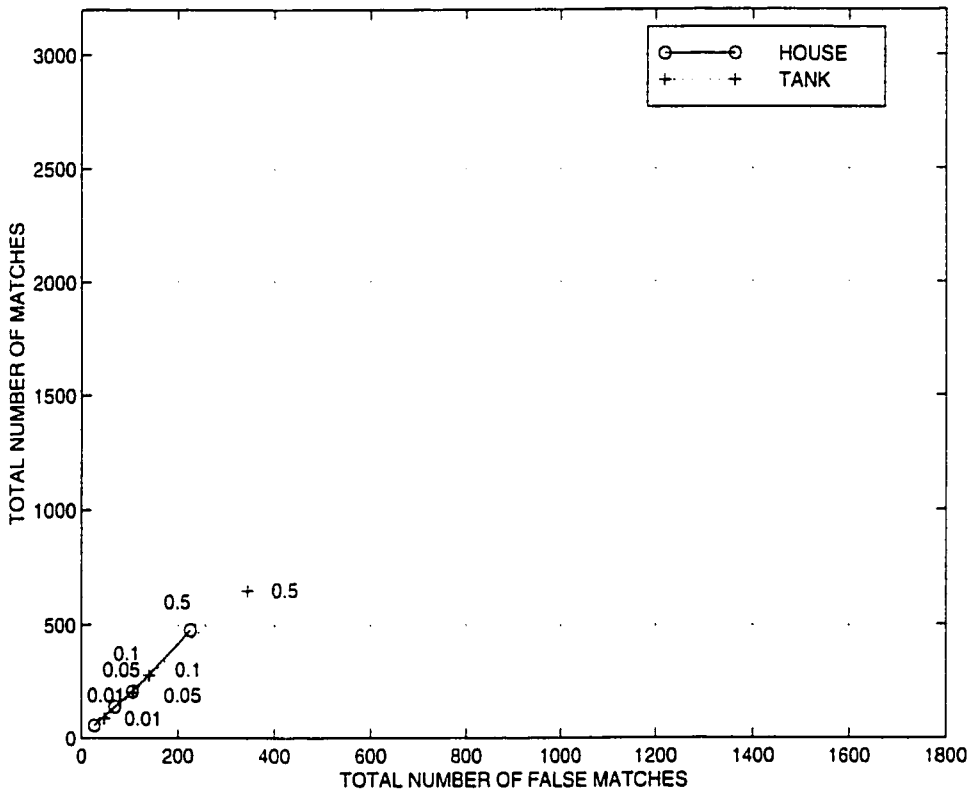


Figure B.7: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and curvature.

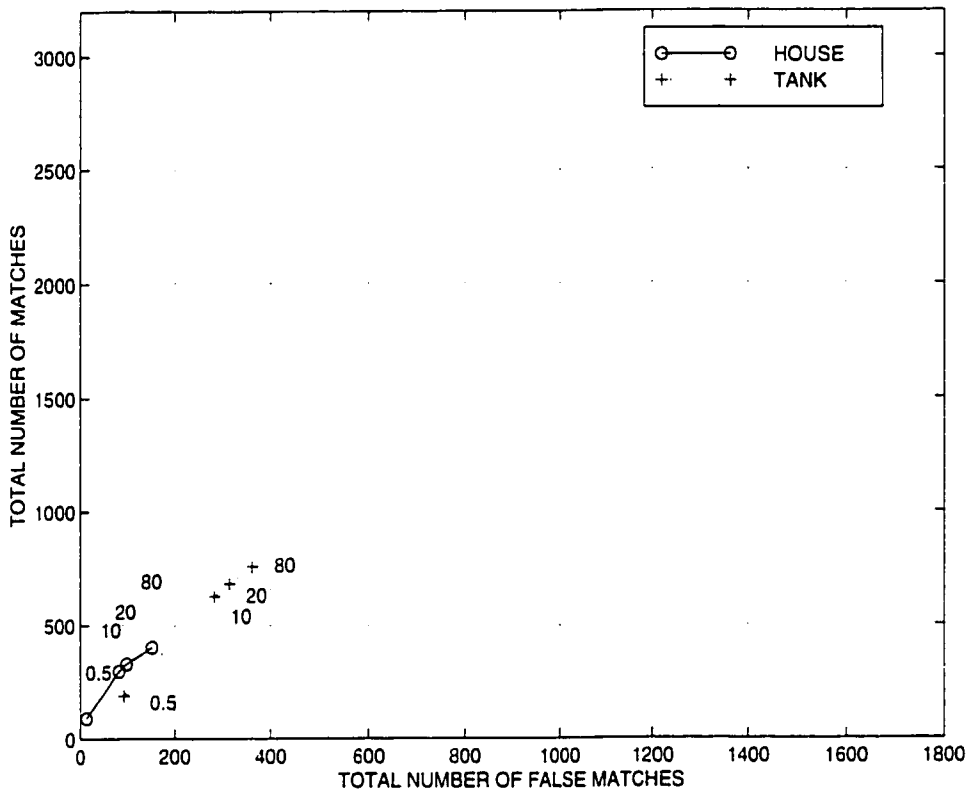


Figure B.8: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and 2DRigidity.

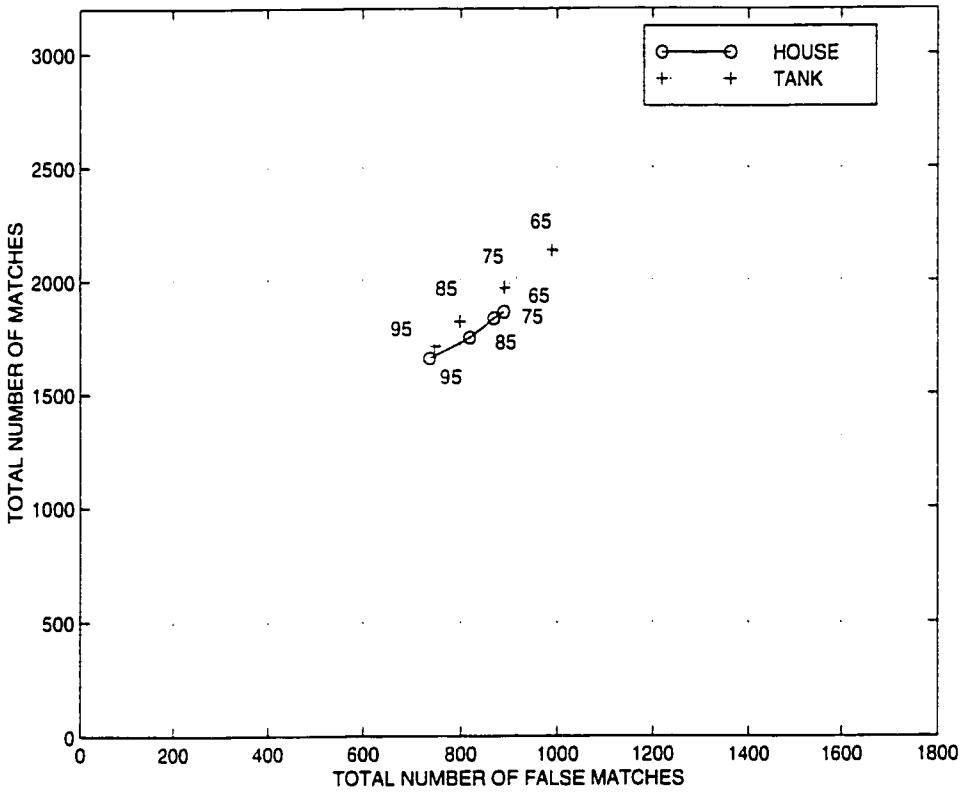


Figure B.9: This graph shows the total number of false matches versus the total number of matches remaining in a *WorkSet* when applying the VNC correlation and figural continuity.

Bibliography

[Ana84] P. Anandan. *Computing Dense Displacement Fields With Confidence Measures In Scenes Containing Occlusion.* 184/SPIE vol. 521 Intelligent Robots and Computer Vision (1984).

[Bak81] H. H. Baker and T. O. Binford. *Depth From Edge and Intensity Based Stereo.* Proc. 7-th International Joint Conf. on Artificial Intelligence, pp. 631-636, Vancouver, Canada, (August 1981).

[Bar80] S. T. Barnard and W. B. Thompson. *Disparity Analysis of Images.* IEEE Trans. PAMI, vol. PAMI-2, pp.333-340, (July 1980).

[Bel96] P. N. Belhumeur. *A Bayesian Approach to binocular stereopsis.* Int'l J. Computer Vision, vol. 19, no. 3, pp. 237-260, (1996).

[Can86] J. Canny. *A computational Approach to Edge Detection.* IEEE Transactions on Pattern analysis and Machine Int. vol. PAMI-8 no. 6 Nov. 1986.

[Dar98] T.Darell. *A radial Cumulative Similarity Transform For Robust Image Correspendence.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 656-662. June 23-25, Santa Barbara, California, (1998).

[Del79] E. J. Delp and O. R. Mitchell. *Image Compression Using Block Truncation Coding.* IEEE Trans. on Communication, vol. COM-27, no. 9, pp, 1335-1342,1979.

[Din98] Dinkar N. Bhat, Shree K. Nayar. *Ordinal Measures for Image Correspondence.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 4, (april 1998).

[Fau96] Olivier Faugeras. *Three-Dimensional Computer Vision 'A Geometry Viewpoint'.* The MIT Press.

[Gra97] Graeme A. Jones. *Constraint, Optimization, and Hierarchy: Reviewing Stereoscopic Correspondence of Complex Features.* Computer Vision and Image Understanding, vol. 65, no. 1, January, pp. 57-78, (1997).

[Gri81] W. E. L. Grimson. *From Images To Surfaces.* MIT Press, Cambridge, (1981).

[Han74] M. J. Hannah. *Computer Matching of Areas in Stereo Images.* Stanford A.I. Memo 239, (July 1974).

[Han84] M. J. Hannah. *Description of SRI's Baseline Stereo System.* Technical Report 342, SRI International, (October 1984).

[Han89] M. J. Hannah. *A System For Digital Stereo Image Matching.* Photogrammetric Engineering and Remote Sensing, 55(12):1765-1770, (December 1989).

[Har88] C. Harris and M. Stephens. *A Combined Corner and Edge Detector.* Proc. 4th Alvey Vision Conf., pp. 147-151 (1988).

[Hof87] William Hoff, Narendra Ahuja. *Extracting Surfaces From Stereo Images An Integrated Approach.* The computer society Of the IEEE, First International Conference On CV, (1987).

[HoM95] Radu Horaud, and Olivier Monga. *Vision Par Ordinateur, Outils Fondamentaux.* Ed. Hermes.

[Hor87] R. Horaud and Th. Skordas. *Stereo Matching Through Feature Grouping and Maximal Cliques.* IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-11(11):1168-1180, (November 1989).

[Hu94] Xiaoping Hu, Narendra Ahuja. *Matching Point Features with Ordered Geometric, Rigidity, and Disparity constraints.* IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 16, no. 10, pp. 1041, (October 1994),.

[Jor91] J.R. Jordan III, A.C. Bovic. *Using a Chromatic Information in Edge-Based Stereo Correspondence.* CVGIP. Image Understandung, vol 54 no.1, July, pp. 98-118, (1991).

[Kim87] Yeon c.Kim, and J.K. Aggarwal. *Positioning Three-Dimensional Objects Using Stereo Images.* IEEE Journal Of Robotics And Automation, Vol. RA-3, no.4, (August 1987).

[Kim86] N. H. Kim and A. C. Bovik. *A Solution To The Stereo Correspondence Problem Using Disparity Smoothness Constraint.* Proc. of IEEE Conf. Systems, Man and Cybernetics, Atlanta, (October 1986).

[Lag98] R. Laganière. *A Morphological Operator For Corner Detection.* Pattern Recognition, vol. 31, no. 11, pp. 1643-1652, 1998.

[Li94] Z.N. Li. *Stereoi Correspondence Based on Line Matching in Hpough Space Using dynamic Programming.* IEEE Trans. Systems, Man Cybernetics, 24(1), (1994).

[Luo93] Q. T. Luong, R. Deriche, O. Faugeras and T. Papadopoulo. *On Determining the Fundamental Matrix: Analysis of Different Mcthods and Experimental Results.* INRIA, report no. 1894 (1993).

[Luo95] A. Luo and H. Burkhardt. *An Intensity-Based Cooperative Bidirectional Stereo Matching With Simultaneous Detection of discontinuities and Occlusions.* Int'l J. Computer Vision, vol. 15, pp. 171=188, (1995).

[Oht85] Y. Ohta and T. Kanade. *Stereo by Intra and Inter-Scanline Search Using Dynamic Programming.* IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-7(2), pp. 139-154, (March 1985).

[Pog85] T. Poggio, V. Torre and C. Koch. *Computational Vision and Regularization Theory.* Nature, vol. 317, pp.314-319, (1985).

[Pla97] F. Pla and J. A. Marchant. *Matching Feature Points in Image Sequences throuh a Region-Based Method.* Computer Vision and Image Understanding, vol.66, no. 3, pp. 271-285 (June 1997).

[Pol85] S.B. Pollard, J.E.W. Mayhew, J.P. Frisby. *PMF a Stereo Correspondence Algorithm Using a Disparity Gradient Limit.* Perception, vol. 14, pp. 449-470, (1985).

[Pra85] K. Pradzy. *Detection of Binocular Disparities.* Bio. Cybern. 52, pp. 93-99, (1985).

[Sko88] Th. Skordas. *Mise en Correspondence et Reconstruction Stereo Utilisant une Description Structurelle des Images.* PhD thesis, Institut National Polytechnique de Grenoble. (October 1988).

[Smi97] S. M. Smith and J. M. Brady. Susan. *A New Approach to Low Level Image Processing.* Int. J. Comput. Vision(1997).

[Tor97] P.H.S Torr and A. Zisserman. *Performance Characterizaton of Fundamental Matrix Estimation Under Image Degradation.* Machine Vision and Applications. vol. 9. pp. 321-333. 1997.

[Vie92] Thierry Vieville and Olivier D. Faugeras *Robust and Fast Computation of Unbiased Intensity Derivatives in Images.* Lecture Notes in CS 588. pp. 203-209. (ECCV'92).

[Wei98] Guo-Qing Wei. W. Brauer. and G. Hirzinger. *Intensity and Gradient-Based Stereo Matching Using Hierarchical Gaussian Basis Functions.* IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 20. no. 11 (November 1998).

[Wen98] Juyang Weng. Narendra Ahuja. and Thomas S.Huang. *Two-View Matching.* Coordinated Science Laboratory. University of Illinois. Urbana. IL61801. IEEE (1998).

[Wen] J. Weng. *A Theory of Image Matching.* Centre de Recherche de Montreal.

[Zha93] Zhengyou Zhang. *Le Probleme de la Mise en Correspondence: L'etat de l'art.* INRIA. report no. 2146. (december 1993).

[Zha94] Zhengyou Zhang. Rachid Deriche, Olivier Faugeras, Quang-Tuan Luong. *A robust Technique for Matching Two Uncalibrated Images Through the*

Recovery of the Unknown Epipolar Geometry. INRIA, report no. 2273, (may 1994).